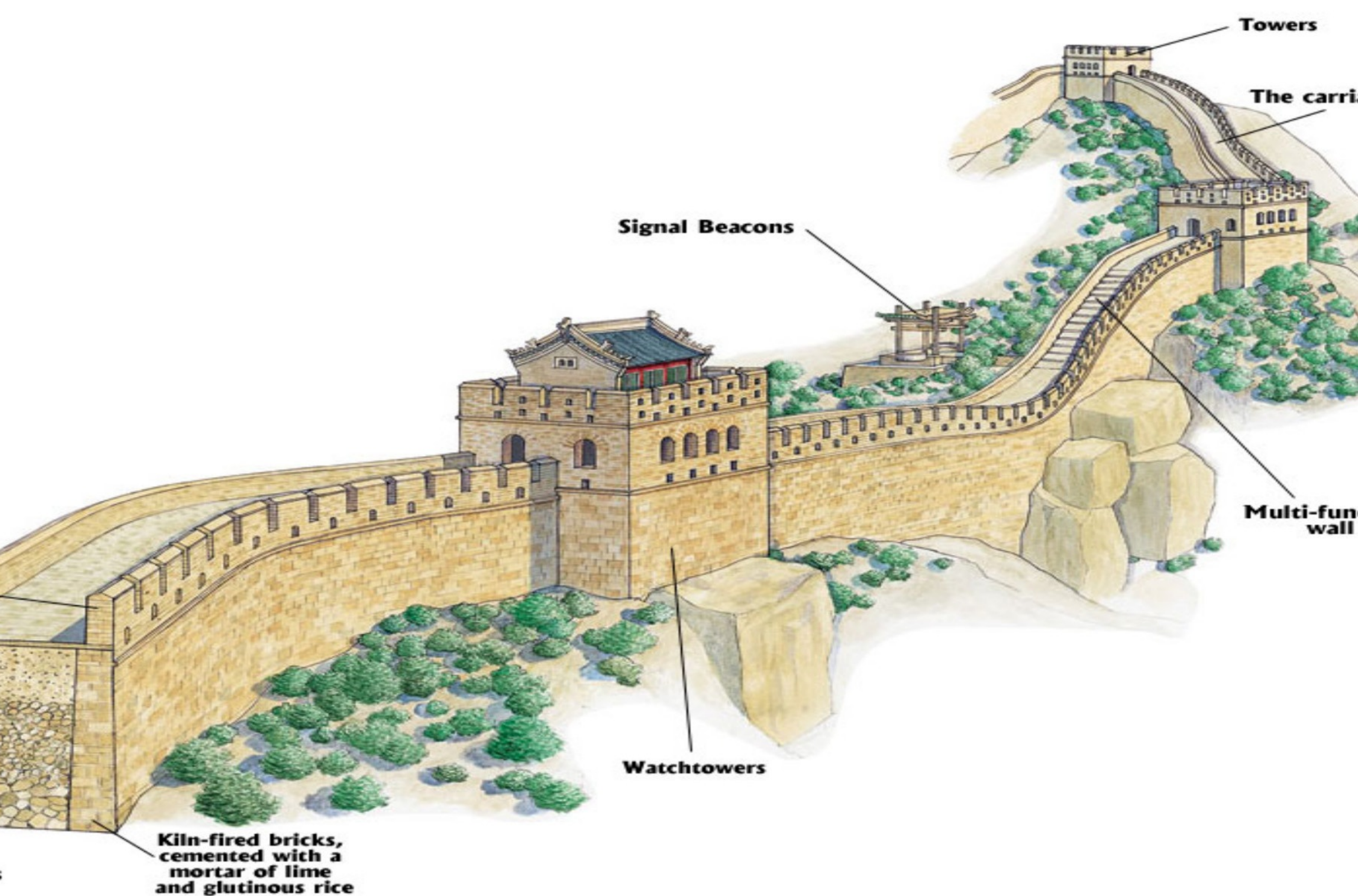
穿越长城



https://github.com/softwaredownload/openwrt-fanqiang

Table of Contents

史上最详细的OpenWrt shadowsocks路由器自动翻墙教程

手把手教你路由器刷OpenWrt固件，自动穿越万里长城。

本教程翻墙方案的特点

放弃建立黑名单的方案吧，被墙的网站每天在大量增加，有限的人生不能在无穷的手工添加黑名单、重启路由器中渡过。

大道至简，一劳永逸！

- 建立国内重要网站名单，在国内进行dns查询
- 其他网站通过 shadowsocks客户端向 shadowsocks服务端进行dns查询
- 国内或亚洲的IP流量走国内通道
- 其他流量通过shadowsocks服务端转发
- 屏蔽ISP劫持相关IP
- 屏蔽国内外的广告

知识若不分享，实在没有意义

2014年6月 Dropbox壮烈被墙。

查资料发现，著名的开源路由器固件OpenWrt支持家里的路由器 TP-Link WR2543N V1 ，于是就给路由器安装了OpenWrt并设置为自动智能翻墙。

自由的感觉真好: youtube, hulu, twitter, facebook, google...

什么是圣人，圣人就是得到和付出比较均衡的人。天地生我，我敬天地；父母育我，我亦养父母；网上获得知识，也要在网上分享知识。于是，花了许多天，查资料，写教程，调试固件，不知不觉一天就过去了。

希望你应用本教程后，也把你的过程写下来，合并到这个项目中来:
https://github.com/softwaredownload/openwrt-fanqiang

如何编译OpenWrt shadowsocks自动翻墙固件

- 首先把本项目clone到本地目录，如 ~/Downloads/openwrt-fanqiang

- 原始配置文件
  - ~/Downloads/openwrt-fanqiang/openwrt/default 默认配置文件夹
  - ~/Downloads/openwrt-fanqiang/openwrt/wndr4300 针对特定路由器型号的配置文件，此处以wndr4300为例
- 复制配置文件
  - 本地建立配置文件目录，如 ~/Downloads/openwrt-wndr4300
  - 复制默认配置文件夹下面的文件到 ~/Downloads/openwrt-wndr4300/ 下
  - 如果有针对特定路由器的配置文件，也复制到~/Downloads/openwrt-wndr4300/，并覆盖同名文件
- 修改配置文件，编译后就直接可以用了。否则刷上固件后登录路由器再修改。主要修改：
  - openwrt-wndr4300/etc/shadowsocks.json
  - openwrt-wndr4300/usr/bin/shadowsocks-firewall
  - openwrt-wndr4300/etc/uci-defaults/defaults
- 编译自定义固件，设置FILES=~/Downloads/openwrt-wndr4300

本项目规定的默认值

```
shadowsocks server:            1.0.9.8
shadowsocks server_port:    1098
shadowsocks local_port:        7654
shadowsocks tunnel_port:    3210
shadowsocks password:        killgfw
shadowsocks method:            aes-256-cfb
root login password:        fanqiang
WIFI password:                icanfly9876   (for DIR-505 and TLWR2543 before 2015: wsjdw,8181)
```

如何使用预编译翻墙固件：

- shadowsocks 服务端按照本项目规定的默认值进行设置（除了server IP)
- 路由器刷OpenWrt shadowsocks翻墙固件
- 登录路由器修改server IP：

```
# Modify 1.0.9.8 to your server IP address
vi /etc/shadowsocks.json

# Modify 1.0.9.8 to your server IP address
vi /usr/bin/shadowsocks-firewall

/etc/init.d/shadowsocks restart
```

- 以上修改测试通过后，建议再修改 shadowsocks password, 路由器root password
- 不建议修改端口号
- 少数时候需要重启路由器才能使修改生效

相关资源

- Netgear WNDR4300 预编译翻墙固件(2015-12-23): https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/
- D-Link DIR-505 预编译翻墙固件(2015-12-24): https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/
- TP-Link TLWR2543 预编译翻墙固件(2015-12-24): https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/
- shadowsocks-libev-polarssl_2.4.3.ar71xx.ipk (2015-12-20): https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/

授权许可

在线阅读OpenWrt翻墙路由器教程:

- https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details
- https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md

无线路由器刷**OpenWrt**固件的准备工作

在给你的路由器刷新固件之前，有必要先了解：

1. 什么是无线路由器固件
2. 准备支持OpenWrt路由器
3. 如何备份路由器配置

最简单的路由器刷**OpenWrt**固件翻墙教程**:**

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读**OpenWrt**翻墙路由器教程**:**

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

什么是无线路由器固件

网络的本质是知识的开放与共享。人类社会进步速度，如果原来是自行车速，加上网络后，就坐上了火箭。

一个热爱学习的人，必然要查找一些英文学习资料，在某个国家的某个阶段必然会遇到一个问题：怎么Google搜索这么烂，经常打不开，YouTube真差劲,加载半天还在打转...

后来，可能会发现，不是人家烂，而是有人故意为之。

怎么办呢？有很多种办法解决这个问题，其中一个较好的方案是从家用无线路由器上解决，然后全部有线和无线设备都可以无障碍上网了。

路由器的原厂固件限制了用户自行开发功能，我们必须要给路由器刷上特定的固件，并进行一些设置才可以翻墙。

无线路由器就好比是一台小电脑。电脑上安装了Windows XP, Windows 7, Windows 8，或者Ubuntu等操作系统就可以使用了。固件就是给路由器使用的操作系统，是固化在路由器芯片内的操作系统。

常用的开源第三方无线路由器固件

1. 开源OpenWRT路由器固件：部署复杂、灵活性高
   这也是本文系列所用的固件。发展成熟，支持的硬件多。
2. 开源DD-WRT路由器固件：支持广泛、功能全面 DD-WRT比较实用，通过网页对固件进行配置的功能强大，但是定制和扩展比较困难。
3. 开源Tomato路由器固件：衍生版本众多 原始版本固件代码自2010年后就再没有更新。

本系列教程使用OpenWrt来讲解路由器翻墙方法。

支持OpenWrt的路由器

现在3G手持设备已经普及，一般情况下读者家里都已经有无线路由器了，到底能不能刷上OpenWrt固件呢？到OpenWrt官方网站查一下就知道了。

打开 支持OpenWrt无线路由器列表 这个页面，搜索一下。比如我家原来的无线路由器型号是TP-LINK TL-WR2543N, 同时按下Ctrl+F, 输入 **WR2543** 就可以找到,如下图：

Model	Version	Status	Target(s)	Platform	CPU Speed (MHz)	Flash (MB)	RAM (MB)	V
TL-WR2543ND	1.0	12.09	ar71xx	Atheros AR7242	400	8	64	A A (c

从上图可以看出，OpenWrt支持 WR2543N 无线路由器版本1。此外，还可以看出更多信息，比如芯片类型是ar71xx, 芯片型号是Atheros AR7242，CPU频率是400 MHz，原厂带8MB Flash, 64MB RAM内存。

目前 WR2543N已经比较少见。如果你购买其他品牌，建议Flash在8 MB或以上, RAM在64MB以上。

如果你准备买新路由器，可以在上面列表中查找OpenWrt推荐路由器型号，能买到的话，再以关键词 型号 **OpenWrt** 在搜索引擎搜索相关信息，确保你想购买的型号能比较容易地刷上OpenWrt固件。

作为新手来说，推荐使用 D-Link DIR-505, 可能是最便宜的适合学习OpenWrt的路由器，如果你的应用场景要求不高，也可以用来作为日常使用的路由器。

怎样备份原厂路由器配置文件

提示，刷机有风险，如果不当操作，或者有其他意外发生，路由器可能变成砖头，本文系列旨在提供参考，刷机风险由读者自负，作者不承担任何责任，也没有义务提供个别指导。

本文作者给 WR2543N 刷 OpenWrt 固件不下10次，因为完全没有经验，有几次刷了后不能进入管理界面，只能用手机3G上网查找解决方案，还好 WR2543N 非常容易进入安全模式，然后重新刷固件，解决了问题。作为初学者，一定要购买容易进入安全模式的路由器。

对于本文作者来说，现在已经不需要原厂固件了，但是在第一次刷OpenWrt前，我还是把原厂固件的配置文件作备份，建议读者也是如此。

怎样备份原厂固件，WR2543N的原厂说明书说得很详细，建议找出来详细阅读。

LAN 和WAN的区别

什么是LAN和WAN,第一次听到这种专业名词容易让人头大。

LAN并不是一个单词，而是三个英文单词的缩写：Local Area Network，查出这三个单词的意思，就比较好理解了，就是 本地区域网络 的意思。本地，比如是室内，公司内，办公室内都是本地，也就是LAN是用来连接本地电脑的。

WAN，Wide Area Network，广泛区域网络，也就是连向更广泛的外部的网络，一般家用就是通向ADSL modem，再通过ADSL modem连接互联网。

路由器通常有多个LAN口，一个WAN口。

在WR2543N路由器的后背，有并排4个的网线插口，叫LAN口，单独的一个网线插口叫WAN口，WAN口旁边还有个USB插口。把ADSL的线插在WAN口。备好一根网线，一头插路由器的任意一个LAN口，另一头插电脑。

设置电脑LAN口IP地址

路由器和电脑都处在本地网络里面，为了互相区分，本地网络的每台设备都需要有不同的IP地址。

本路由器默认 LAN 口 IP 地址是 192.168.1.1， 默认子网掩码是 255.255.255.0

电脑的IP地址要和路由器的不同，我们可以设置电脑的本地IP地址为动态获取。如果手动设置IP地址，那么计算机IP地址必须为192.168.1.X （X）是2到254之间的任意整数），子网掩码须设置为255.255.255.0，默认网关须设置为192.168.1.1

以Windows XP 系统为例，介绍计算机参数的设置步骤。

右键单击桌面上的 网上邻居 图标，选择 属性，在打开的 网络连接页面中，右键单击“本地连接”，选择状态，打开“本地连接状态”进行操作。详细步骤请见购机时附带的手册。

登录路由器管理界面

打开网页浏览器，在浏览器的地址栏中输入路由器的 IP地址：192.168.1.1，可以看到下图:



所示登录界面，输入用户名和密码（用户名和密码的出厂默认值均为admin），单击确定按钮。

备份原厂路由器固件配置文件

登录路由器管理界面后，选择菜单，系统工具→备份和载入配置，可以在如下图所示备份或载入路由器配置文件。

配置备份功能可以将路由器的设置以文件形式保存到电脑中，以备下次使用；在升级路由器软件或在载入新的配置文件前备份路由器的原有配置，可以有效防止升级软件或载入新配置文件过程中丢失原有配置的问题。

配置载入功能则可以将先前保存的或已编辑好的配置文件重新载入。

备份和载入配置文件

您可以在这保存您的设置。我们建议您在修改配置及升级软件前备份您的配置文件。

备份配置文件

您可以通过载入配置文件来恢复您的设置。

路径： 浏览... 载入配置文件

注意：
1、导入配置文件后，设备中原有的用户配置将会丢失。如果您载入的配置文件有误，可能会导致设备无法被管理。
2、载入配置文件的过程不能关闭路由器电源，否则将导致路由器损坏而无法使用。载入过程约20秒，当载入结束后，路由器将会自动重新启动。

怎样刷第三方路由器固件OpenWrt

经过前面的准备，终于要给亲自给路由器刷OpenWrt固件了。有可能失败，有可能成功。一连嘴里念叨FGW (=fuck great wall),一边给自己打气。

有几个问题有必要提一下：

OpenWrt有必要装中文管理界面吗?

我认为不需要。网上最新最全面的信息都是英文的。GFW在不断进步，我们也要不停地学习。我们要感谢GFW，让我们每天多记几个单词。一些步骤的操作，我特意截图并加上了步骤标识，实在记不住就每次打开这个教程照着图示来。

在开源的Linux类操作系统里连接OpenWrt进行操作

我认为有必要从现在开始切换到Linux类操作系统了。Windows已经开始走向没落，开源操作系统渐渐赶上闭源商业操作系统。

为什么呢？随着技术的不断进化，开源的技术合作越来越方便。我打个比方，如果佛教老大释珈牟尼，基督教创始人耶稣，不开源恐怕也会穷途末路。

再说OpenWrt就是微型的Linux操作系统，熟悉了Linux，学习OpenWrt就很容易了。

在以后的教程里，都是在Ubuntu下对OpenWrt进行管理。如果有两台电脑，建议一台装Ubuntu，如果只有一台电脑，可以装Ubuntu和Windows双启动。

最简单的路由器刷OpenWrt固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读OpenWrt翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

怎样下载适合自己路由器的OpenWrt固件

下载最新版的OpenWRT固件

- 进入OpenWrt固件下载主页面：

  http://downloads.openwrt.org/

  Binary Releasesy就是最后的稳定发行版，如目前是

  ```
  Chaos Calmer 15.05
  Released: Fri, 11 Sep 2015
  ```

  Development Snapshots是开发版，包含最新的功能，但可能不够稳定。

  http://downloads.openwrt.org/snapshots/trunk/

  如果使用Snapshots没有什么问题，当然是最好的选择，否则可以尝试一下稳定发行版。

  下面以snapshots，和WR2543举例。

- 选择CPU类型

  打开页面后，选择你的路由器的芯片型号进入，很多是ar71xx系列，于是进入了：

  http://downloads.openwrt.org/snapshots/trunk/ar71xx/

- 选择 Flash类型

  再选择Flash类型，比如WR2543是generic，网件WNDR4300路由器是nand。

  http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/

  再选择你的路由器型号，页面搜索 wr2543，找到了吗。有两个文字供下载，一个文件结尾是 factory.bin,适合原厂固件下刷，另一个文件名结尾是sysupgrade.bin,适合已经是OpenWrt系统下刷。

OpenWrt官方wiki下载OpenWrt固件 for WR2543

OpenWrt官方网页上有WR2543N的专页,详细介绍了刷机步骤及注意事项.

打开官方Wiki页面 TP-Link TL-WR2543ND

上面列出了支持的版本: v1.0和v1.2。我的路由器是v1.0的，可以刷，你的版本如果不是这两个，不能确保能刷成功。

这两个固件都带LuCI 网页管理界面。有时候，如果你升级了不带LuCI的固件，命令行方式又无法搞定OpenWRT上网参数设置,就需要先在电脑里下载带LuCI的固件，scp复制到路由器升级，再通过网页设置。

有两个固件供下载:

- openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-factory.bin - Installing OpenWRT from factory
- openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-sysupgrade.bin - Upgrading an existing OpenWRT install

一定要注意:

- 在原厂固件上刷OpenWrt, 要用固件文件名带 **factory** 的.bin文件.
- 已经刷了OpenWrt固件, 再升级 OpenWrt固件时就要用文件名带 **sysupgrade** 的 .bin文件.

现在我们是在原厂固件基础上刷 OpenWrt, 自然是下载第一个文件,也就是 openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-factory.bin

要确保下载下来的文件完整,下载过程没有中断,如果下载下来的文件不完整,并把这个不完整的文件刷进机器,恢复起来很麻烦,有可能变砖.

下载预编译的翻墙固件 for WR2543

- https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/

通过网页界面WR2543刷OpenWrt教程

通过有线或无线连上WR2543路由器

打开浏览器,输入路由器的IP地址: 192.168.1.1

回车,在密码验证框,输入用户名: **admin** 密码也是 **admin**

路由器固件升级

选择菜单系统工具→ 软件升级

软件升级

通过升级本路由器的软件，您将获得新的功能。

文件： 浏览...

当前软件版本： 3.13.12 Build 120109 Rel.35902n

当前硬件版本： WR2543N v1 00000000

注意： 升级时请选择与当前硬件版本一致的软件。 升级过程不能关闭路由器电源，否则将导致路由器损坏而无法使用。升级过程约20秒，当升级结束后，路由器将会自动重新启动。

升 级

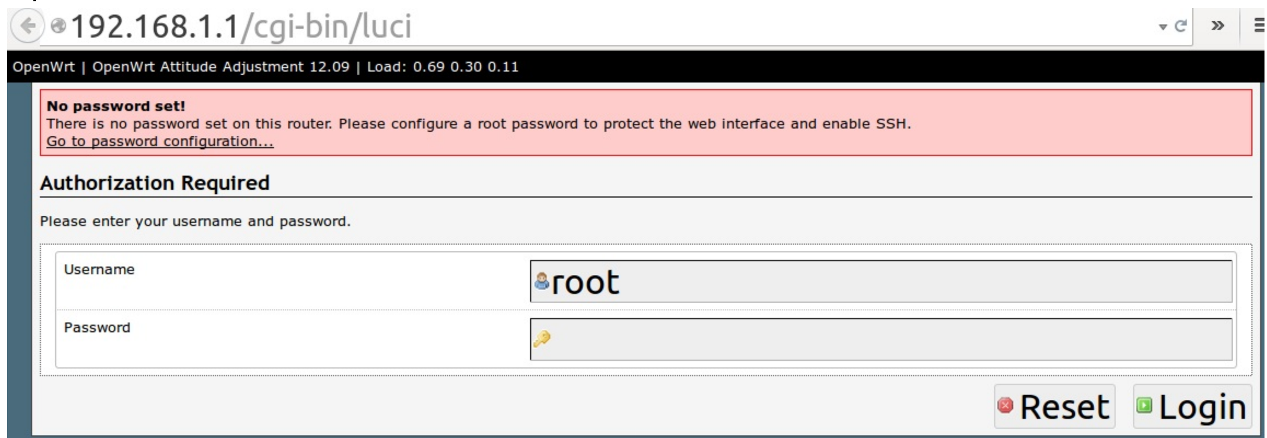
点击 浏览 按钮选择下载的文件 **openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-factory.bin**

注意，文件名必须是...factory.bin。

再单击 升级 进行软件升级。要注意，在刷固件过程中不可停电或其他原因造成中断，否则路由器就变砖了。

等待几分钟

等锁形的指示灯不闪了，在浏览器输入地址: 192.168.1.1 回车,如果正常的话,就进入了OpenWrt 的LuCI网页管理界面了.



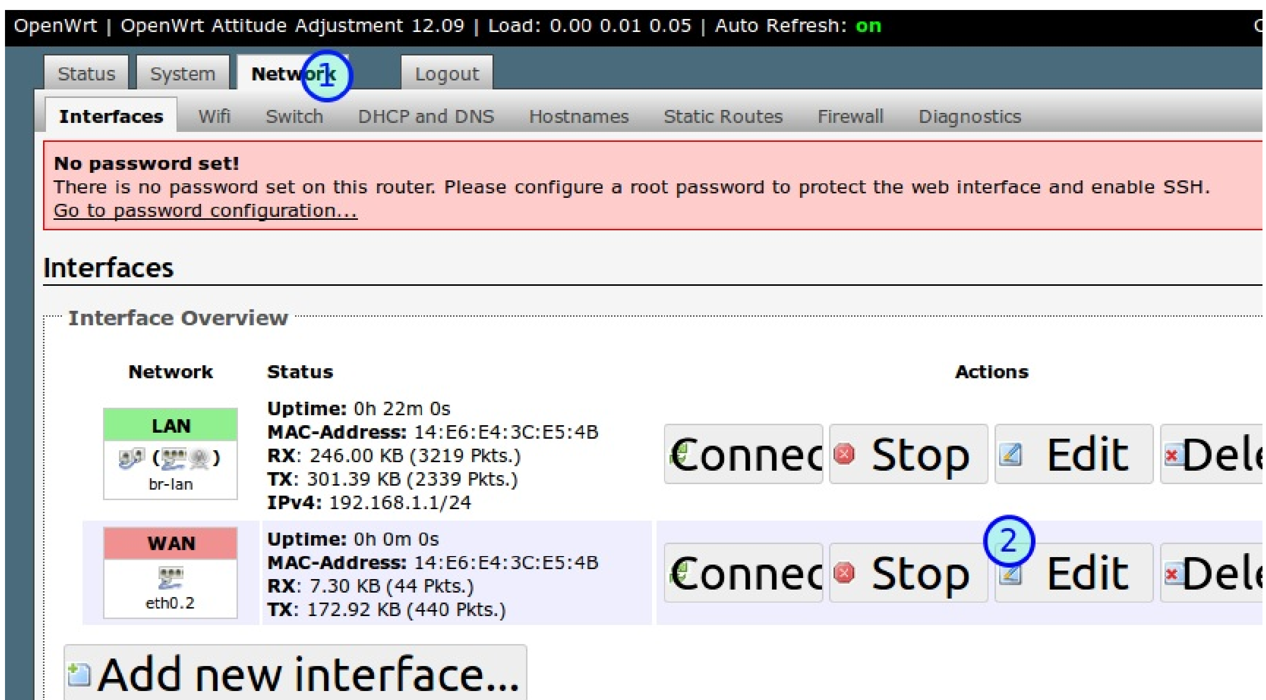
默认用户名是root,默认密码是空。点 **Login** 直接登录。

网页界面OpenWrt拨号上网设置教程

见面界面登录路由器后，就可以设置上网参数了。

编辑OpenWrt WAN设置

选择上面的 Network，在 Interface里，WAN右边，选择Edit。WAN和ADSL moderm相连，设置拨号上网自然是在WAN而不是LAN.



配置OpenWrt拨号上网密码

进去后，在协议 Protocol 下拉列框里，选择拨号上网的协议，也就是 PPPoE, 再点击下面的 Switch Protocol切换协议。

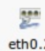
Interfaces - WAN

On this page you can configure the network interfaces. You can bridge several interfaces by ticking the "bridge interfaces" field and enter the names of several network interfaces separated by spaces. You can also use VLAN notation INTERFACE.VLANNR (e.g.: eth0.1).

Common Configuration

General Setup

Status


eth0.2

Uptime: 0h 0m 0s
MAC-Address: 14:E6:E4:3C:E5:4B
RX: 9.96 KB (60 Pkts.)
TX: 231.08 KB (588 Pkts.)

Protocol: PPPoE ①

Really switch protocol?  Switch protocol ②

 Reset  Save  Save & App

OpenWrt拨号上网用户名密码设置

1. PAP/CHAP username: 拨号上网用户名
2. PAP/CHAP password:　拨号上网密码
3. 点击 **Save & Apply** 保存并应用设置

Common Configuration

General Setup | Advanced Settings | Physical Settings | Firewall Settings

Status


pppoe-wan

RX: 0.00 B (0 Pkts.)
TX: 0.00 B (0 Pkts.)

Protocol: PPPoE

PAP/CHAP username: admin ①

PAP/CHAP password: •••••••• ②


Access Concentrator: auto
 Leave empty to autodetect

Service Name: auto
Leave empty to autodetect

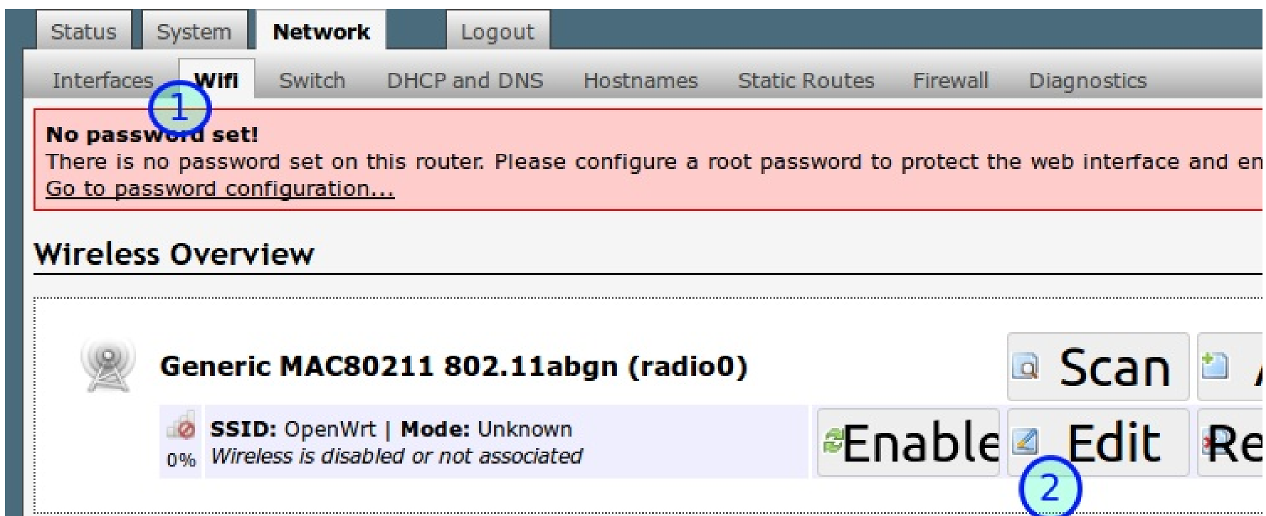
③
 Reset  Save  Save & Apply

这时，连接LAN的电脑应该已经可以上网了，但无线设备还不行。

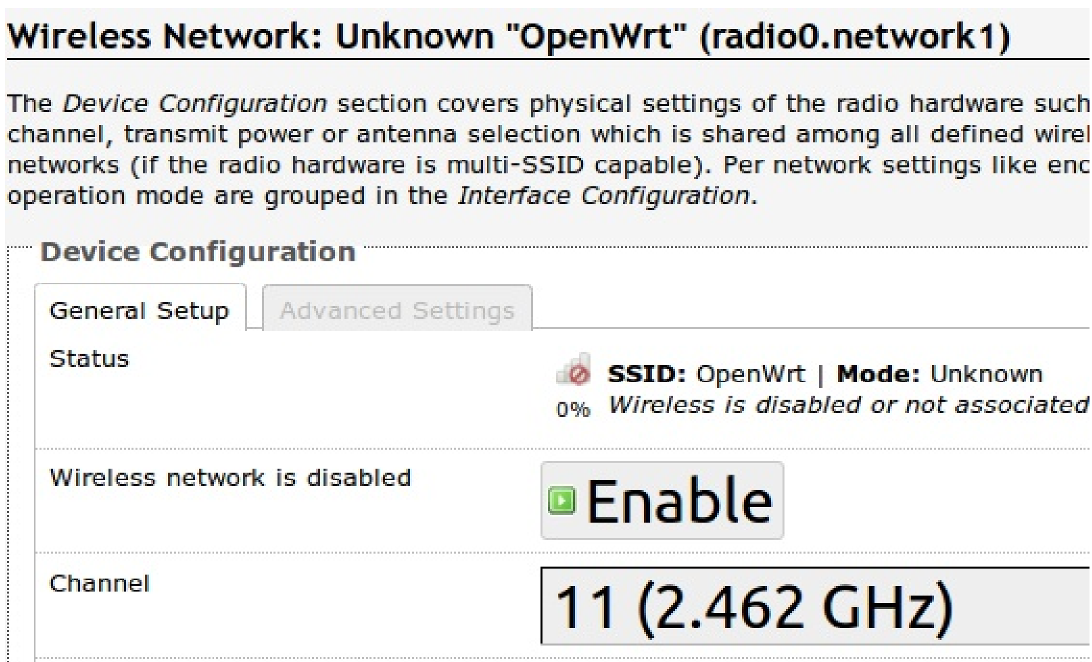
网页界面OpenWrt无线(Wifi)设置图文教程

登录OpenWrt路由器后：

选择 Network, Wifi, Edit



点击Enable按钮，这时无线设备已经可以连上Wifi



默认ESSID就是OpenWrt，没有密码。不想做活雷锋的加个密码吧。

OpenWrt Wifi密码设置

把ESSID改成 eastking-wr2543,然后：

- 点击Wireless Security进入OpenWrt无线安全设置
- Encryption加密方式，WPA2-PSK
- Key密码：killgfw
- Save & Apply　保存并应用设置

Transmit Power | 27 dBm (501 mW) | dBm

Interface Configuration

General Setup | Wireless Security | MAC-Filter

Encryption | WPA2-PSK

Cipher | auto

Key | ••••••••

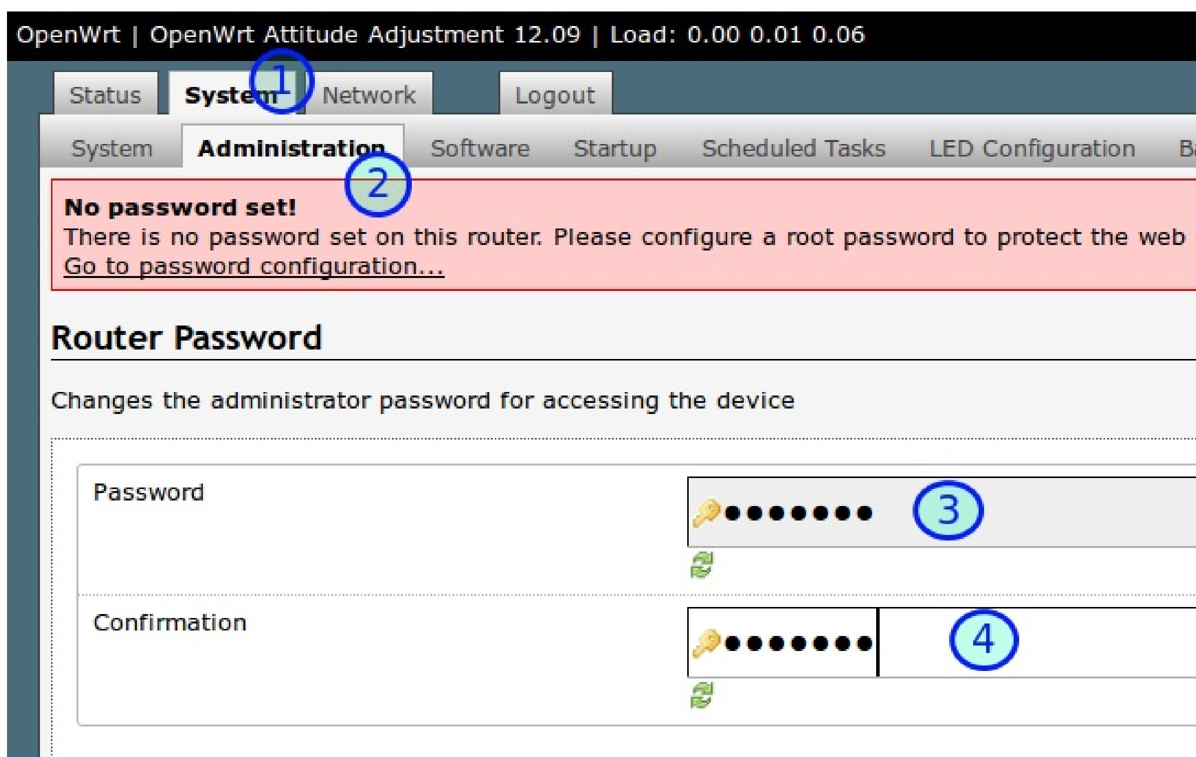
Reset　Save　Save & Apply

这时，所有无线设备都可以通过OpenWrt路由器上网了。

OpenWrt管理界面登录密码设置

你注意到没有， 网页上方有一个红色的框框(No password set!)一直在提示我们： 小人不得不防，**OpenWrt**叫你设一个路由器管理界面登录密码呢!

1. 点击最上面的System进入系统设置
2. 再点击Administration进入管理员设置
3. 密码Password: fanqiang
4. 确认密码Confirmation: fanqiang



5. 其他设置：下面的： Gateway ports，勾选 **Allow remote hosts to connect to local SSH forwarded ports**（允许远程主机连接本地SSH转发端口）,这样我们就可以用SSH命令行的方式管理路由器。最后点击右下角 Save & Apply保存并应用设置。

网页界面怎样备份OpenWrt设置

现在有线和无线上网都正常了。应该把现有的OpenWrt设置备份一下，因为我们还要经常折腾OpenWrt，有时一个设置错误，可能就上不了网，有了备份，就可以快速恢复。

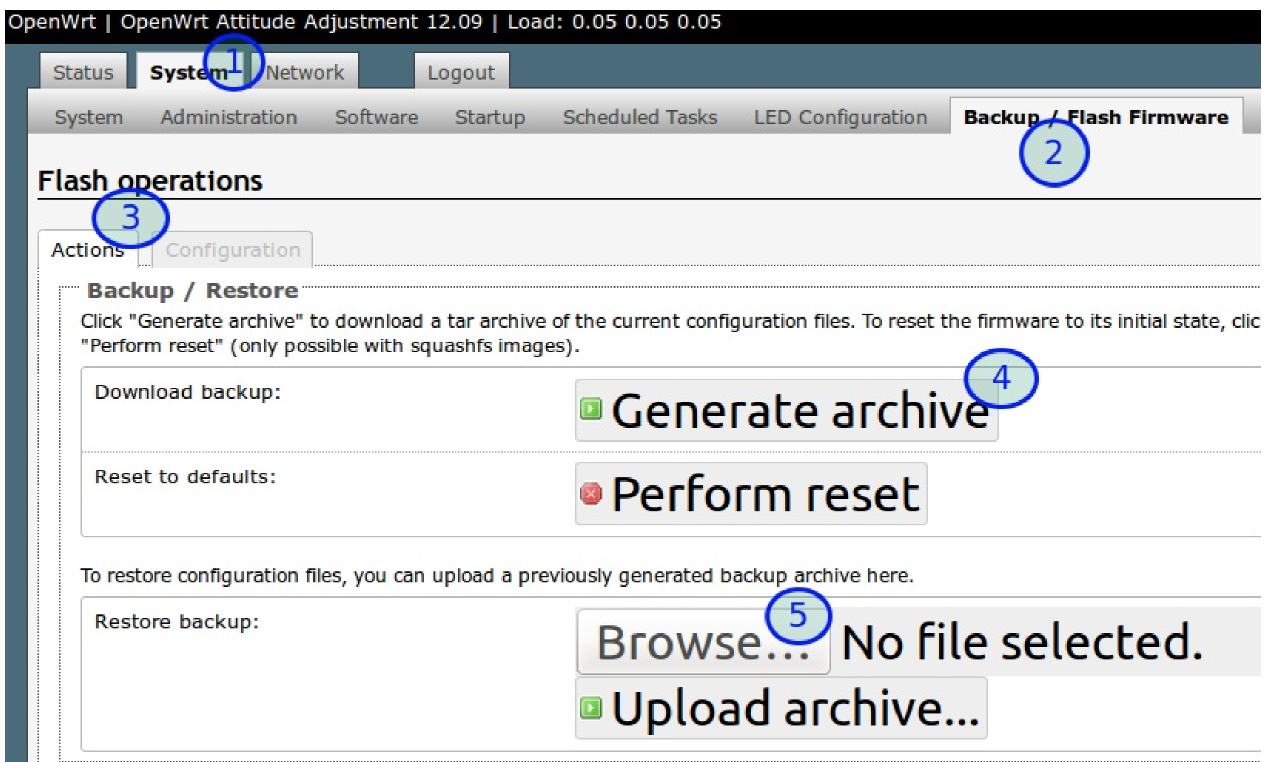
选择System系统设置

选择Backup / Flash Firmware备份恢复固件

Actions动作

Generate生成备份文件并保存到电脑

如果以后你要恢复备份，就点击Browse...浏览并选择先前备份的文件来恢复



通过LuCI网页界面升级OpenWrt固件

我们现在已经给TP-Link WR2543N刷上了OpenWrt固件，并且可以正常上网了。如果要升级OpenWrt固件，又该怎么做呢？

有两个途径升级固件：

- LuCI web界面升级
- SSH命令行登录路由器升级

本节就讲 web界面升级固件的方法。

下载OpenWrt升级用固件sysupgrade.bin

下载用于WR2543N路由器的升级固件，升级用固件文件名中有sysupgrade字样。

还是到OpenWrt Wiki页面 TP-Link TL-WR2543ND

点击 openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-sysupgrade.bin 下载。

其实这个固件的核心和我们先前安装的...factory.bin一样，我们是出于实验目的，演示升级固件的方法，

用前文讲过的方法从网页登录OpenWrt路由器

开始升级OpenWrt固件

1. System系统
2. Backup / Flash Firmware备份或刷新固件
3. Flash new firmware, Browse...选择我们刚下载下来的固件
4. Flash image...刷新固件
   注：如果Keep settings保持勾选，升级固件后，原来的设置就会保留，不用重新设置拨号上网参数。

Status | System ① | Network | Logout

System | Administration | Software | Startup | Scheduled Tasks | LED Configuration | **Backup / Flash Firmware** ② | Reboot

Flash operations

Actions | Configuration

Backup / Restore

Click "Generate archive" to download a tar archive of the current configuration files. To reset the firmware to its initial state, click "Perform res images).

Download backup: Generate archive

Reset to defaults: Perform reset

To restore configuration files, you can upload a previously generated backup archive here.

Restore backup: Browse... No file selected.

Upload archive...

Flash new firmware image

Upload a sysupgrade-compatible image here to replace the running firmware. Check "Keep settings" to retain the current configuration (req image).

Keep settings: ☑

Image: Browse... ③ openwrt-ar71xx-

Flash image... ④

怎样进入OpenWrt 安全恢复模式(WR2543N为例)

有时候，我们可能操作失误，无法进入LuCI网页界面管理恢复固件，这时就需要进入安全模式来恢复了。

不同的路由器，进入安全模式的方法可能有所差别，本文系列适用于 TP-LINK WR2543N。

安全模式是玩OpenWrt的救命仙丹。能熟练进入安全模式来恢复设置，是OpenWrt已经上手的一个标志。

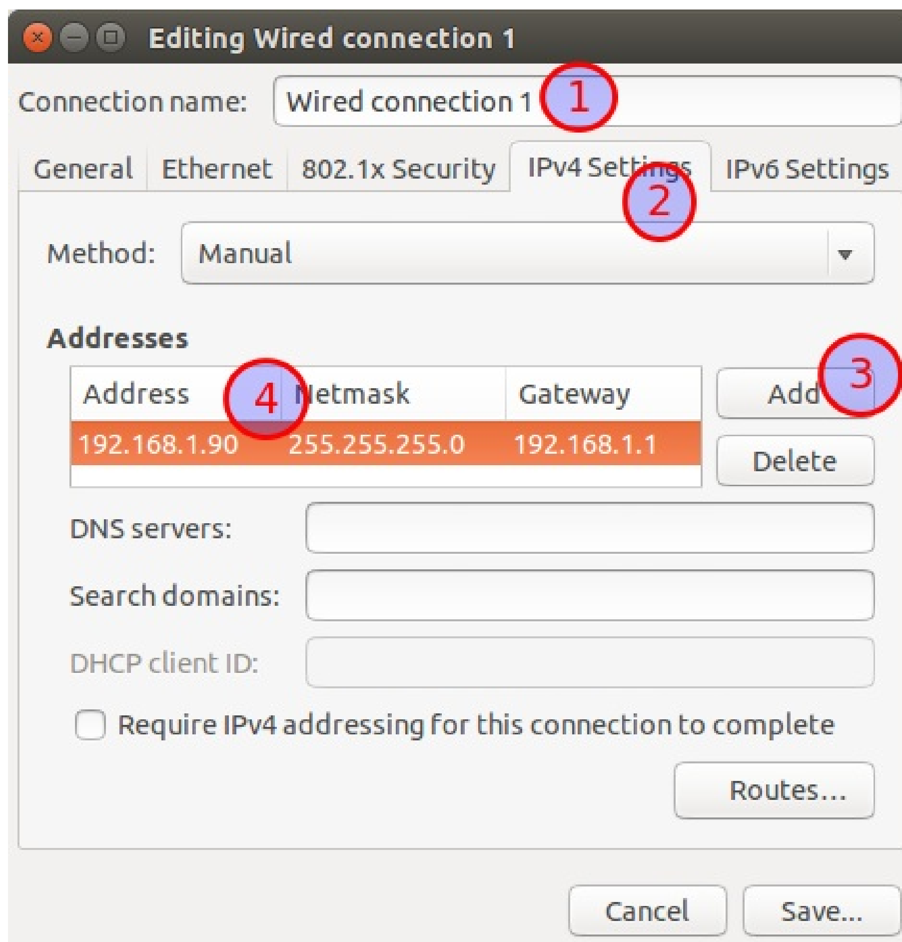
进入安全模式时，没有无线连接可用，所以我们要有线的方式登录OpenWrt。OpenWrt默认的IP地址是192.168.1.1，我们要设置电脑有线连接的IP地址类似于192.168.1.x, 其中x是2至255的数字。

WR2543N无线路由器进入OpenWrt安全模式的方法：

1. 用网线把路由器和电脑连接起来，设置电脑网卡的IPv4地址

   以Ubuntu为例，点击桌面右上角连接符号，选择 **Edit Connections**, 再选择 Ethernet连接，点击　Edit 按钮，在弹出的窗口中选择 IPv4 Settings, Method选择Manual，Address栏点击Add，设置如下：

   - Address: 192.168.1.90
   - Netmask: 255.255.255.0
   - Gateway: 192.168.1.1



2. 在Ubuntu运行命令：

```
sudo tcpdump -Ani eth0 port 4919 and udp
```

3. 重启路由器,当WR2543N的锁形指示灯刚一开始闪烁时，立即按路由器背面的wps按钮3次

4. Ubuntu命令行界面出现：

> Please press button now to enter failsafe

```
donald@Software-Download:/$ sudo tcpdump -Ani eth0 port 4919 and udp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:08:41.122133 IP 192.168.1.1.46561 > 192.168.1.255.4919: UDP, length 1001
E.....@.@.............7..!...Please press button now to enter failsafe.......
.............................................................................
.............................................................................
.............................................................................
.............................................................................
.............................................................................
```

5. Ubuntu命令行执行(有时可以不需tcpdump直接telnet)：

```
telnet 192.168.1.1
```

这时就成功登录了OpenWrt，如下图:

```
donald@Software-Download:/$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

 === IMPORTANT ============================
  Use 'passwd' to set your login password
  this will disable telnet and enable SSH
 ------------------------------------------

  github.com/softwaredownload/openwrt-fanqiang

BusyBox v1.19.4 (2013-03-14 11:28:31 UTC) built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 ATTITUDE ADJUSTMENT (12.09, r36088)
 -----------------------------------------------------
  * 1/4 oz Vodka      Pour all ingredients into mixing
  * 1/4 oz Gin        tin with ice, strain into glass.
  * 1/4 oz Amaretto
  * 1/4 oz Triple sec
  * 1/4 oz Peach schnapps
  * 1/4 oz Sour mix
  * 1 splash Cranberry juice
 -----------------------------------------------------
root@(none):/#
```

1. 设置登录OpenWrt SSH登录密码:

```
passwd
#输入密码 fanqiang
```

如果出现：

```
passwd: /etc/passwd: Read-only file system
passwd: can't update passwd file /etc/passwd
```

就输入 `mount_root` 再重新passwd设置管理员密码。如下图：

```
 --------------------------------------------------
root@(none):/# passwd
Changing password for root
New password:
Retype password:
passwd: /etc/passwd: Read-only file system
passwd: can't update password file /etc/passwd
root@(none):/# mount_root
switching to jffs2
root@(none):/# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@(none):/#
```

telnet登录路由器后，可以用vi命令修改设置。

这时如果你试图用浏览器登录192.168.1.1进入管理界面的话，可能失败。

重启路由器，路由器锁形指示灯先是慢闪，到变成常亮时，你又可以登录 192.168.1.1管理界面。一切恢复正常。

OpenWrt sysupgrade 命令行更新固件到最新版

下面我们要使用 sysupgrade 更新固件到snapshot最新版。

要注意的是，trunk包含试验的功能，可能不稳定，刷机风险自己承担。

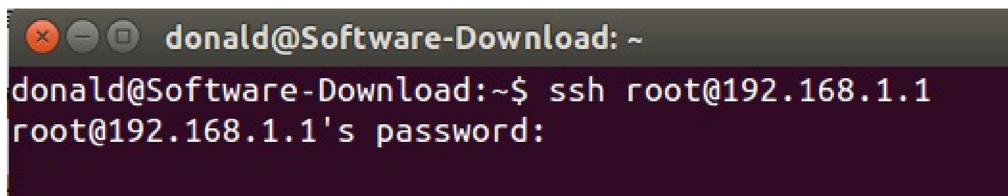
在浏览器里登录 192.168.1.1 进行固件升级是比较简单的。今天我们要尝试的的是命令行刷机升级。命令行的方式更强大。

SSH登录路由器

在Ubuntu里，按Ctrl+Alt+T打开命令行终端，输入：

```
ssh root@192.168.1.1
```

输入密码，登录成功。



进入OpenWrt /tmp目录

```
cd /tmp
```

检查OpenWrt路由器是否有足够的内存

```
df -h
```

可以看出， **/tmp** 还有29.5MB可用空间，而升级固件在3MB左右，足够了。

```
root@OpenWrt:~# cd /tmp
root@OpenWrt:/tmp# df -h
Filesystem                Size      Used Available Use% Mounted on
rootfs                    5.1M    284.0K      4.8M   5% /
/dev/root                 2.0M      2.0M         0 100% /rom
tmpfs                    30.1M    608.0K     29.5M   2% /tmp
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/mtdblock3            5.1M    284.0K      4.8M   5% /overlay
overlayfs:/overlay        5.1M    284.0K      4.8M   5% /
```

下载OpenWrt最新trunk版本固件

1. 在Ubuntu里浏览器打开 http://downloads.openwrt.org/snapshots/trunk/
2. TP-LINK WR2543N路由器的芯片类型是ar71xx，就点击 ar71xx 目录进入。要注意，路由器的芯片类型千万不能搞错，不同路由器很可能是不同的。



3. TP-LINK WR2543路由器的Flash类型为 generic，于是进入了 http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/
4. 按Ctrl+F查找自己的路由器型号。比如我输入的是 **wr2543**, 有两个固件，升级用的是 **sysupgrade.bin**文件。右键点击该链接，复制下载地址。在FireFox里是 **Copy Link Location**复制链接地址。
5. 回到Ubuntu命令行终端， 下载固件到 **/tmp** 目录。TP-LINK wr2543路由器是这样的:

```
 root@OpenWrt:/tmp# wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

md5校验，确保下载的固件完整:

```
root@OpenWrt:/tmp# wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/generi
c/md5sums
root@OpenWrt:/tmp# md5sum -c md5sums 2> /dev/null | grep OK
openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin: OK
```

输出结尾是OK，说明固件是完整的。

OpenWrt sysupgrade命令升级OpenWrt固件

```
root@OpenWrt:/tmp# sysupgrade -v openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysu
pgrade.bin
...
Upgrade completed
Rebooting system...
```

过约2分钟，等路由器重启成功，如果没有意外，会发现有线和无线上网都正常。但浏览器192.168.1.1无法登录，因为snapshots版本固件是不带LuCI网页管理界面的。没有也好，可以节省路由器的存储空间，也可以学习一下命令行管理OpenWrt路由器。

参考：

- generic.sysupgrade
- sysupgrade source code
- sysupgrade doc

命令行 uci设置 OpenWrt router模式上网参数

如果路由器可以正常上网的前提，我们可以ssh登录路由器，直接在路由器的/tmp目录wget下载最新版固件并sysupgrade命令进行固件升级。

有时候，路由器无法上网，这时候，可以在电脑里下载好固件，再把固件复制到路由器，再sysupgrade升级或设置其他参数。

只要能进入路由器的安全模式，并ssh登录路由器，一切都不是问题。

Ubuntu下载OpenWrt for TP-LINK wr2543N trunk版固件

```
cd ~/Downloads
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

scp复制固件到OpenWrt路由器 /tmp目录

```
scp openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

ssh登录OpenWrt路由器

```
ssh root@192.168.1.1
cd /tmp
```

sysupgrade升级固件并取消保留原来配置文件

注意，升级后将无法上网，也没有LuCI网页界面可以设置，必须以命令行方式设置好上网参数。

如果在下面的实验中，命令行方式无法搞定路由器上网，就只能在电脑里下载好带luCI的固件，scp复制固件到路由器升级固件，然后以网页方式设置上网。

在进行这一步前，确保你熟练掌握以前部分教程。

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgra
de.bin
```

参数 `-n` 表示升级时不保留原来的配置文件。固件刷好后会自动重启，这时要用前文教程讲过的方法进入OpenWrt安全模式，登录路由器并重新设置root密码。

下面假设你已经登录了路由器并设好了root密码。

OpenWrt uci命令行设置拨号上网：

```
root@OpenWrt: uci set network.wan.proto='pppoe'
root@OpenWrt: uci set network.wan.username='wan-username'
root@OpenWrt: uci set network.wan.password='wan-password'
root@OpenWrt: uci set network.wan.peerdns=0
```

wan-username替换成你自己的拨号上网用户名，wan-password替换成你自己的密码。

OpenWrt uci命令行设置无线上网：

```
root@OpenWrt: uci set wireless.@wifi-device[0].channel=11
root@OpenWrt: uci set wireless.@wifi-device[0].txpower=17
root@OpenWrt: uci set wireless.@wifi-device[0].disabled=0
root@OpenWrt: uci set wireless.@wifi-device[0].country='CN'
root@OpenWrt: uci set wireless.@wifi-iface[0].mode='ap'
root@OpenWrt: uci set wireless.@wifi-iface[0].ssid='eastking-tlwr2543'
root@OpenWrt: uci set wireless.@wifi-iface[0].encryption='psk2'
root@OpenWrt: uci set wireless.@wifi-iface[0].key='icanfly9876'
```

uci设置说明：

- channel 信道
- txpower 功率
- disabled 是否启用无线，0表示启用
- ssid 名称，推荐后面以路由器型号结尾，这样调试多个路由器时不会混淆。
- encryption 加密方式
- key 无线密码，如果你照上文的设置不动，好处是忘记密码时可以上 http://www.github.com/softwaredownload/openwrt-fanqiang 来查看。

允许远程主机用ssh的方式登录路由器及设置时区

```
root@OpenWrt: uci set dropbear.@dropbear[0].GatewayPorts='on'
root@OpenWrt: uci set system.@system[0].zonename='Asia/Shanghai'
root@OpenWrt: uci set system.@system[0].timezone='CST-8'
```

ssh登录OpenWrt相关高级设置（你可能暂时用不到)

```
root@OpenWrt: uci set dropbear.@dropbear[0].Port=22
root@OpenWrt: uci set dropbear.@dropbear[0].PasswordAuth=off
root@OpenWrt: uci set dropbear.@dropbear[0].RootPasswordAuth=off
```

说明（不懂千万别乱设）：

- Port ssh默认端口就是22,可以改成其他的提高安全性
- PasswordAuth ssh是否启用密钥登录。如果你改成off，又没有设置好ssh私钥和安装好LuCI，你将无法ssh方式登录路由器，唯一的办法就是安全恢复模式登录重新开始设置。
- RootPasswordAuth 是否允许root用密码登录，如果已经设置好了ssh私钥就可以改成off增加安全性。

启用新的网络和无线设置

```
root@OpenWrt: /etc/init.d/dropbear restart
root@OpenWrt: /etc/init.d/system restart
root@OpenWrt: /etc/init.d/network restart
```

怎么样，有线和无线上网又都回来了吧！

注意，有的人在网上贴出了他的完整配置文件/etc/config/network 和/etc/config/wireless，如果你复制他的文件覆盖你的文件，再修改用户名和密码，可能会出问题，因为不同路由器的硬件配置可能不同。

命令行 uci设置OpenWrt ap模式上网参数

前面章节已经说过了router模式上网的设置方法，主要是设置OpenWrt路由器wan口的拨号上网参数。

OpenWrt路由器工作在ap模式下时，自身不需要拨号上网了，设置稍有不同。

什么时候需要用到ap模式上网

光纤包月或包年上网时，通信公司一般会给一个猫，如果猫里没有设置自动拨号上网，那么OpenWrt路由器就要用router模式，我们自己手动在OpenWrt里设置拨号上网。

如果猫里已经设置好拨号上网，从猫的lan拉出一根网线插到电脑的网线接口，电脑直接可以上网了，这时OpenWrt路由器就要设置成ap模式。

如果是公司里或者家里有多个路由器，上级路由器里拉出一根网线插到电脑网线接口，电脑直接可以上网了，这时OpenWrt路由器作为下级路由器，需要设置成ap模式。

OpenWrt路由器AP模式网络设置

从光猫或上级路由器拉出网线，插到OpenWrt路由器的lan口(注意不是插到wan口)

命令行登录OpenWrt路由器，设置参数

假设光猫或上级路由器的IP地址是192.168.1.1，我们设置OpenWrt路由器的lan地址是192.168.1.254，这也是登录OpenWrt路由器的的地址。

```
uci set network.lan.gateway=192.168.1.1
uci set network.lan.dns=192.168.1.1
uci set network.lan.ipaddr=192.168.1.254

uci set network.wan.proto=none

uci commit network

uci set dhcp.lan.ignore=1
uci commit dhcp

uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='icanfly9876'

uci commit wireless
wifi

/etc/init.d/network restart
```

客户端连接OpenWrt路由器

如果不需要翻墙，客户端连上OpenWrt路由器后，直接就可以上网了。

如果上级路由器没有翻墙，客户端需要通过OpenWrt路由器翻墙，客户端连接到OpenWrt路由器后，按照下面设置：

- 设置客户端连接的的IPv4地址是 192.168.1.6（最后的6不和其他设备的地址相同即可）
- 设置子网掩码为255.255.255.0
- Router(网关)和DNS设为路由器lan口的地址，此处为192.168.1.254

原理：以OpenWrt路由器作为DNS服务器，我们已经把OpenWrt设置成翻墙路由器，连上的客户端自然就可以打败功夫网了。

Auto-Join

IPV4 ADDRESS

Configure IP	Manual >
IP Address	192.168.1.6
Subnet Mask	255.255.255.0
Router	192.168.1.254

DNS

Configure DNS　Manual >

iPhone连接ap模式的翻墙路由器，IPV4设置如上图

Back Configure DNS Save

Automatic

Manual ✓

DNS SERVERS

192.168.1.254

Add Server

iPhone连接ap模式的翻墙路由器，DNS设置如上图

给初始不具备翻墙能力的路由器配置软件源

当我们刚给路由器刷上OPENWRT后，其只具备基本的上网功能。 这里如果我们使用opkg update安装软件时，发现其根本不能update,因为它要连接的download.operwrt.org 本身也在被墙列表中。（或未被墙，但速度奇慢。）

方法有几种

方法1，为opkg配置代理

1. 在路由配置文件中，为openwrt的opkg配置代理。 来源openwrt wiki

   进入路由器菜单，选择系统-软件，点击“配置”页签，输入：

   option http_proxy http://proxy.example.org:8080/

   option ftp_proxy ftp://proxy.example.org:2121/

2. 也可以直接vi /etc/opkg.conf去修改。

   option http_proxy http://proxy.example.org:8080/

   option ftp_proxy ftp://proxy.example.org:2121/

方法2， 架设一个不需代理就可访问的软件源。

1. 建立OPENWRT的镜象文件。

   到

   https://downloads.openwrt.org

   下载你所需要的芯片对应的文件夹内的目录及文件拖到本地。 比如我的是：ar71xx/nand/packages/ 下的base luci management 等文件夹。使用wget命令可以整站拖。

   ```
   wget -m -np https://downloads.openwrt.org/chaos_calmer/15.05/ar71xx/nand/packages/
   ```

   如果是旧版的路由，请自行到找到对应的版本去下载，比如

   ```
   wget -m -np https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/nand/packages/
   ```

2. 使用任一建站软件，比如Nignx建立简单的http服务器.

3. 将opkg-config内的软件源改成自己的http服务器

```
- src/gz chaos_calmer_base http://192.168.1.121:8008/base

- src/gz chaos_calmer_luci http://192.168.1.121:8008/luci

- src/gz chaos_calmer_packages http://192.168.1.121:8008/packages

- src/gz chaos_calmer_routing http://192.168.1.121:8008/routing

- src/gz chaos_calmer_telephony http://192.168.1.121:8008/telephony

- src/gz chaos_calmer_management http://192.168.1.121:8008/management
```

Done.

OpenWrt＋shadowsocks-libev实现路由器自动翻墙

相信经过前面的教程，大家对OpenWrt和Linux Ubuntu有一定的熟悉了。如果还不熟悉Ubuntu，就安装Ubuntu，实际使用一个月。

前面的文章都是技术准备，有基础的读者可以略过。在本章中，我们要OpenWrt路由器安装shadowsocks-libev来实践翻墙。

最简单的路由器刷**OpenWrt**固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读**OpenWrt**翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

什么是shadowsocks-libev翻墙软件

shadowsocks-libev 是一个 shadowsocks 协议的轻量级实现，是 shadowsocks-android, shadowsocks-ios 以及 shadowsocks-openwrt 的上游项目。其具有以下特点：

1. 体积小巧。静态编译并打包后只有 100 KB。
2. 高并发。基于 libev 实现的异步 I/O，以及基于线程池的异步 DNS，同时连接数可上万。
3. 低资源占用。几乎不占用 CPU 资源，服务器端内存占用一般在 3MB 左右。
4. 跨平台。适用于所有常见硬件平台，已测试通过的包括 x86，ARM 和 MIPS。也适用于大部分 POSIX 的操作系统或平台，包括 Linux，OS X 和 Cygwin 等。
5. 协议及配置兼容。完全兼容 shadowsocks 协议，且兼容标准实现中的 JSON 风格配置文件，可与任意实现的 shadowsocks 客户端或服务端搭配使用。

shadowsocks-libev 包括服务端和客户端两部分，一共三个模块。

1. ss-server：服务器端，部署在远程服务器，提供 shadowsocks 服务。
2. ss-local：客户端，提供本地 socks5 协议代理。
3. ss-redir：客户端，提供本地透明代理，需要与 NAT 配合使用
4. ss-tunnel: 客户端，本地端口转发

官网地址：

https://github.com/shadowsocks/shadowsocks-libev

翻墙软件Shadowsocks-libev服务端设置

要利用 shadowsocks-libev翻墙，首先要有一台国外的服务器安装并运行shadowsocks 服务端。如果还没有服务器,可以到 digitalocean 购买一台SSD虚拟服务器VPS。

Ubuntu安装 shadowsocks-libev服务端

for Debian 9("Stretch"), unstable, Ubuntu 16.10 and later derivatives:

```
sudo apt-get update
sudo apt-get install shadowsocks-libev
```

for other versions:

```
#Add GPG public key:
wget -O- http://shadowsocks.org/debian/1D27208A.gpg | sudo apt-key add -

# Ubuntu 14.04 or above
sudo add-apt-repository "deb http://shadowsocks.org/ubuntu trusty main"

# Debian Wheezy, Ubuntu 12.04 or any distribution with libssl > 1.0.1
sudo add-apt-repository "deb http://shadowsocks.org/debian wheezy main"

sudo apt-get update
sudo apt-get install shadowsocks-libev
```

Ubuntu 16.10上确认shadows-libev已经运行：

```
sudo systemctl status shadowsocks-libev
```

上述命令的效果：

- 安装ss-local ss-redir ss-server ss-tunnel...到 /usr/bin
- 启动文件　/etc/init.d/shadowsocks-libev
- 配置文件 /etc/shadowsocks-libev/config.json (旧版是/etc/shadowsocks/config.json)
- 一些默认启动配置 /etc/default/shadowsocks-libev （旧版是/etc/default/shadowsocks）

编辑shadowsocks-libev配置文件

```
sudo vi /etc/shadowsocks-libev/config.json
```

改成类似如下：

```
{
    "server":"1.0.9.8",
    "server_port":1098,
    "password":"killgfw",
    "method":"aes-256-cfb"
}
```

必须把server `1.0.9.8` 改成你自己的，或者改成 `0.0.0.0` 表示监听本机，其他可以不改。如果访问变慢，时断时续，这可能是受到干扰了，可以尝试改变加密方式，如改成bf-cfb。

防火墙 ufw 设置

如果你已经启用了防火墙 ufw，那么要用如下命令开放server_port，注意把下面的1098换成你的实际端口:

```
sudo ufw allow 1098
```

ubuntu上查看 ufw 是否已经启用：

```
sudo systemctl status ufw
```

控制shadowsocks-libev的方法

```
sudo service shadowsocks-libev start
sudo service shadowsocks-libev stop

在Ubuntu 16.10上安装shadows-libev后，默认已经随机启动了

#设置随机启动 Shadowsocks-libev. 新版已经不需要手动设置随机启动
#sudo update-rc.d shadowsocks-libev defaults 99

#取消shadowsocks-libev随机启动
#sudo update-rc.d -f shadowsocks-libev remove
```

查看ss-server是否已经启动并且带有 -u启动参数

```
ps ax | grep ss-server
```

如果启动正常，返回结果类似如下：

```
/usr/bin/ss-server -c /etc/shadowsocks-libev/config.json -a root -u -f /var/run/shadow
socks-libev/shadowsocks-libev.pid
```

注意其中有-u。如果shadowsocks客户端启用了udp relay, 而服务端启动时不带-u参数，翻墙自然就失败了。

启动shadowsocks-libev服务端

```
sudo service shadowsocks-libev start
```

参考：

- shadows-libev on github
- create a simple linux daemon

OpenWrt路由器运行shadowsocks-libev客户端

shadowsocks-libev for OpenWrt要和OpenWrt一致，否则可能无法安装，或者安装了不能启动。

下面地址可以下载：

http://sourceforge.net/projects/openwrt-dist/files/shadowsocks-libev/

请自行测试是否适用于你的OpenWrt。

如果自行编译翻墙固件，最好按照教程自己编译 shadowsocks-libev for OpenWrt。

shadowsocks-libev选择OpenSSL版还是PolarSSL版

根据依赖的 SSL 库可分为 OpenSSL 和 PolarSSL 两种版本OpenSSL 版依赖 libopenssl, 支持加密方式多, 体积大 PolarSSL 版依赖 libpolarssl, 体积小, 加密方式少。

如果内存大就选OpenSSL版，反之则选PolarSSL版。

安装shadowsocks-libev客户端到OpenWrt路由器（星号替换成实际的字符）

```
~/Downloads$ scp shadowsocks-libev-polarssl_*_ar71xx.ipk root@192.168.1.1:/tmp/
~/Downloads$ ssh root@192.168.1.1
root@OpenWrt:~# cd /tmp
root@OpenWrt:~# opkg install shadowsocks-libev-polarssl_1.*.*_ar71xx.ipk
```

修改shadowsocks-libev客户端配置

```
root@OpenWrt:~# vi /etc/shadowsocks.json
```

改成类似如下：

```
{
    "server":"1.0.9.8",
    "server_port":1098,
    "local_port":7654,
    "password":"killgfw",
    "method": "aes-256-cfb"
}
```

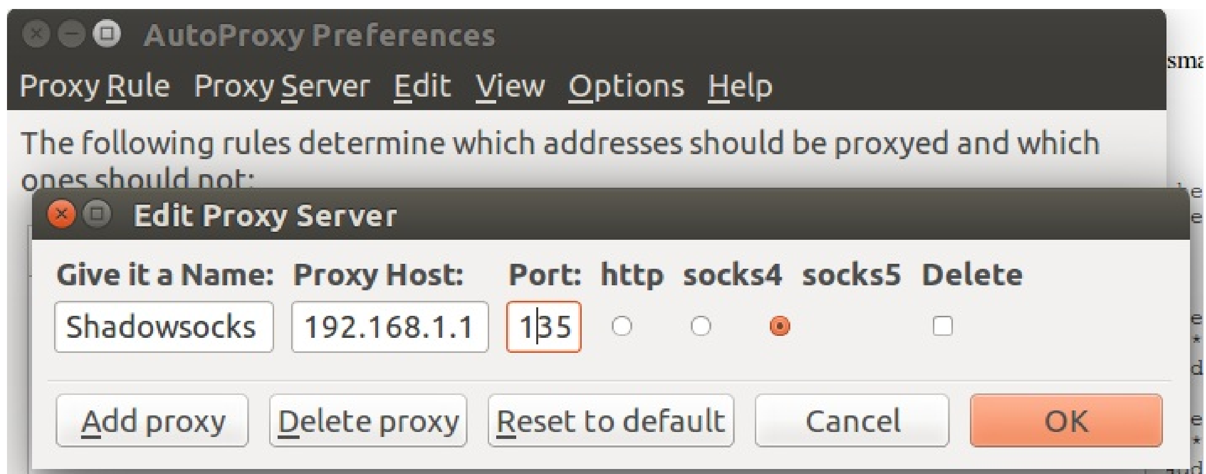
注意，server IP必须修改你的实际IP。其他可以保持默认。

shadowsocks代理上网测试

- 启动shadowsocks 客户端:

  root@OpenWrt:~# ss-local -c /etc/shadowsocks.json

- Ubuntu浏览器代理上网设置，以FireFox配合AutoProxy为例，增加Proxy Server, Proxy Host填192.168.1.1,Port是7654， 勾选Sock5.如下图：



Ubuntu设置AutoProxy的默认代理是shadowsocks,就可以打开被墙的网站如YouTube.com

以前我在每台电脑上都运行一个shadowsocks客户端，每台电脑都要像上面这样配置浏览器代理上网翻墙。

现在路由器里安装了shadowsocks，所有有线和无线上网设备都不用分别安装shadowsocks了，非常方便了。但是还是太复杂，如果家里有十台上网设备，所有要连国外网站的软件都可能要配置代理访问，有些软件还根本没有设置代理的接口。有没有更简单的方法呢？

史上最通俗易懂的OpenWrt翻墙路由器解释

什么是域名和IP地址

每个网站都可以有两个唯一标识：域名和IP地址。域名相当于人的名字，IP地址相当于该人使用的电话号码。（不同之处：域名是唯一的，人的名字会有重名）

网站为什么要有两个标识？域名是为了方便人类记忆的，比如YouTube.com，而电脑处理却喜欢处理数字，纯数字格式的IP地址就是为了让电脑查找计算方便些。

通过域名查询IP的那些事情

我们在浏览器地址栏里输入 www.youtube.com 并回车，到底会发生哪些不可思议的事情呢：

1. 浏览器问就近的某台电脑（叫域名服务器）：Hi, youtube.com的IP地址是什么？
2. 域名服务器：不就是 74.125.239.98
3. 浏览器：谢谢。我就到你给我的地址去找内容了

还有种情况，浏览器第一次问的域名服务器不知道某域名的IP地址：

1. 浏览器问就近的域名服务器：Hi, youtube.com的IP地址是什么？
2. 域名服务器：这个我不知道哇，我帮你问问我的上线
3. 上线服务器：我也不知道哇，我也只好问我的上线，等等，别挂掉
4. 某域名服务器：这么简单还来问我，不就是 74.125.239.98
5. 浏览器：谢谢。我就到你给我的地址 74.125.239.98 去找内容

白脸很忙，不看YouTube(看不懂？)

在中国，YouTube为什么被封？YouTube有几千万，上亿个视频，如果某几个视频让某些人看了不爽，就来个宁可错杀百万，不可放过一个，把整个YouTube给封了，全国人民都无法正常访问YouTube了。

这个时候，又发生了哪些不可告人的事情呢？

1. 浏览器问就近的域名服务器：喂, youtube.com的IP地址是什么？
2. 中国的某域名服务器：这我知道，44.44.44.44，（心里嘀咕，我给你的是太平洋海底的地址，你能找到内容才怪呢，白脸（领导）很忙，天朝很好，访问这种破网站干啥，满屏洋文，我怎么看得懂，哼）

3. 浏览器：谢谢。我这就去找主人需要的内容。。。找了好久，还是什么也没找到,我的命怎么这么苦。。。

阳光底下，每时每刻每秒，这样龌龊的事情在发生千次，万次，亿次...

白脸不知道TCP?

人民要学习新知识，不能容忍这样的事情，于是想到了一些办法， 国内的域名服务器要说谎，那我直接就问国外的服务器比如google的8.8.8.8或8.8.4.4，人家才不会这么卑鄙无耻下流无底线。直接问google的域名服务器的办法存活了一段时间，后来白脸（领导不晒太阳的）又知道了，又不高兴了，于是google的域名服务器IP地址也被封了。

网民的力量是无穷的，有人又发现了，查询域名IP地址的方式有udp和tcp两种，领导暂时只知道udp，我用tcp的方式去问国外域名服务器，就可以得到网站域名对应的真实IP了。

写这篇文章的时候,tcp的方式依然有效。但我怀疑，白难迟早又会不高兴。8.8.8.8或8.8.4.4树大招风，而且白脸最容易记住的就是8和4这两个数字，白脸要发，白脸怕死。所以，我们索性不用8和4了，我们用加密的方式(shadowsockss ss-tunnel)向自己的服务器查询，领导这下该满意了吧！

太阳要升起，网民要雄起

但是，还有问题没有解决：

网站有两种，国内的和国外的。如果不分国内国外全部都到国外去查询域名的IP，访问国内的网站就会变慢。虽然有心逃离，还是无法割断哪。

有几种解决方案：

1. 建国外重要网站名单，简称外单（黑名单，gfwlist），外单上的域名都到国外去查询IP，其他就在国内查询。

   如果IP地址在外单上，就加密访问，领导不知道我访问了这个地址，这样领导的心情可能会好些。

2. 同样是建立外单。不同的是，我不想花费精力去区分某个IP是不是在外单上，IP地址可能经常在变，这样做不怕累吗。我的办法是，不是中国的IP，全部加密访问。

3. 每个人的用途不同，谁有本事建立通用的外单？

   即使有人建立了包含很多域名的外单，网站内容往往是互相引用的，某外单上网站引用了不在外单上的被封网站，导致打网站贼慢，这个该怎么办？难道要手动查看网页源代码，一个个地搜索查找，逐个测试？

最简单有交的方法，是给国内重要网站建立名单，简称内单。内单上的网站都在国内dns，其他网站全部到国外dns。访问非中国的IP都流量加密。

我曾经用过第一种方案，试图用网友整理的外单(ChinaDNS)，但是，在实际使用过程中，经常需要临时增加外单域名并重启路由器，有时一天要重复好多次，不胜其烦.

第三种方案，就是本教程使用的方案，是目前来说比较好的方案。

OpenWrt翻墙路由器内部发生的故事(千万别告诉白脸)：

1. 浏览器：喂，谁知道YouTube.com的IP，主人要用
2. 路由器：稍等，我查下主人设置的内单，稍等。。。不在内单，我通过秘密通道查
3. 浏览器：喂，告诉我baidu.com的IP
4. 路由器：哇，内单，马上就给你
5. 浏览器：请给我IP地址60.188.5.6的内容
6. 路由器：等下，立即就好。。。中国IP，该那就那去取内容。不是中国IP，借道主人的秘密通道去取内容

配置OpenWrt shadowsocks路由器智能自动翻墙

OpenWrt路由器用dnsmasq转发国内重要域名查询

OpenWrt默认自带dnsmasq，我们只要配置一下就好了。ssh登录OpenWrt路由器后:

- 建立dnsmasq.d目录：

```
root@OpenWrt:~# mkdir /etc/dnsmasq.d
root@OpenWrt:~# echo "conf-dir=/etc/dnsmasq.d" >> /etc/dnsmasq.conf
```

- OpenWrt安装GNU wget以支持https下载，下载国内重要网站名单，用国内域名服务器查询iP地址

```
root@OpenWrt:~# cd /etc/dnsmasq.d
root@OpenWrt:/etc/dnsmasq.d# opkg install wget
root@OpenWrt:/etc/dnsmasq.d# wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/accelerated-domains.china.conf
root@OpenWrt:/etc/dnsmasq.d# wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/bogus-nxdomain.china.conf
```

注：accelerated-domains.china.conf 文件中的条目举例：

```
server=/10010.com/114.114.114.114
server=/115.com/114.114.114.114
```

意思是，访问10010.com这个结尾的域名时，dnsmasq会转发到国内的域名服务器114.114.114.114进行dns查询。

gfwlist.conf: 其他域名，转发到shdowsocks-libev ss-tunnel指定的端口dns查询

```
root@OpenWrt:/etc/dnsmasq.d# echo "server=/#/127.0.0.1#3210" > gfwlist.conf
```

上面#是通配符，代表泛匹配所有域名。dnsmasq匹配域名的特点是详细特征优先匹配，因此会先匹配accelerated-domains.china.conf上的域名，如果不匹配，再匹配这条规则：转发到本地端口3210进行域名查询。

后面我们会配置shdowsocks-libev的本地客户端ss-tunnel转发本地端口3210的查询到远程自建服务器。

配置shdowsocks本地客户端ss-redir启动和停止函数

```
root@OpenWrt:/etc/dnsmasq.d# vi /etc/init.d/shadowsocks
```

/etc/init.d/shadowsocks:

```
#!/bin/sh /etc/rc.common

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2015-12

START=95

SERVICE_USE_PID=1
SERVICE_WRITE_PID=1
SERVICE_DAEMONIZE=1

start() {
    sed -i 's/114.114.114.114/127.0.0.1#3210/' /etc/dnsmasq.d/gfwlist.conf
    /etc/init.d/dnsmasq restart

    service_start /usr/bin/ss-redir -b 0.0.0.0 -c /etc/shadowsocks.json -f /var/run/sh
adowsocks.pid -u
    service_start /usr/bin/ss-tunnel -b 0.0.0.0 -c /etc/shadowsocks.json -l 3210 -L 8.
8.8.8:53 -u
    /usr/bin/shadowsocks-firewall
}

stop() {
    sed -i 's/127.0.0.1#3210/114.114.114.114/' /etc/dnsmasq.d/gfwlist.conf
    /etc/init.d/dnsmasq restart

    service_stop /usr/bin/ss-redir
    service_stop /usr/bin/ss-tunnel
    killall ss-redir
    killall ss-tunnel
    /etc/init.d/firewall restart
}
```

shadowsocks本地客户端配置文件start stop函数说明：

- **sed -i 's/127.0.0.1#3210/114.114.114.114/' /etc/dnsmasq.d/gfwlist.conf**
  停止shadowsocks翻墙服务时,要把泛匹配域名的解析转发到国内的dns服务器,这里是114
- **sed -i 's/114.114.114.114/127.0.0.1#3210/' /etc/dnsmasq.d/gfwlist.conf**
  开启翻墙服务时，如果以前停止过shadowsocks翻墙服务,要把泛匹配域名的解析改成通过ss-tunnel 3210端口转发
- **service_start /usr/bin/ss-tunnel -b 0.0.0.0 -c /etc/shadowsocks.json -l 3210 -L 8.8.8.8:53 -u**
  监听本地3210端口，转发到自己的服务器的53端口向8.8.8.8查询DNS
- **/usr/bin/shadowsocks-firewall**
  dnsmasq只是负责域名查询分配转发，查询到IP地址后，是否需要通过shdowsocks加密请求内容，要在shdowsocks-firewall里进行设置
- 运行 `/etc/init.d/shadowsocks stop` 有时并没有结束ss-redir 或ss-tunnel进程，这会导致

修改 `shadowsocks.conf` 后需要重启路由器才能生效。加上 `killall` 强制杀掉进程避免重启。(2016-01-19)
（注：即使加了 killall，有时还是不能杀掉进程，这种情况就只能重启路由器了。也就是说，修改了翻墙配置，有时必须重启路由器才能生效）

- 据tefiszx的建议，ss-redir 新增 -u参数，即翻墙数据走 udp协议。udp速度快，但可能丢包。 tcp速度慢，但不容易丢包。请大家测试（20170711）

配置iptables防火墙转发IP和端口

```
root@OpenWrt:~# cd /usr/bin
root@OpenWrt:~# touch shadowsocks-firewall
root@OpenWrt:~# chmod +x shadowsocks-firewall
root@OpenWrt:~# vi shadowsocks-firewall
```

/usr/bin/shdowsocks-firewall:

```
#!/bin/sh

# Author:    https://github.com/softwaredownload/openwrt-fanqiang
# Date:        2015-12-23

#create a new chain named SHADOWSOCKS
iptables -t nat -N SHADOWSOCKS
iptables -t nat -N SHADOWSOCKS_WHITELIST

# Ignore your shadowsocks server's addresses
# It's very IMPORTANT, just be careful.
iptables -t nat -A SHADOWSOCKS -d 1.0.9.8 -j RETURN
#iptables -t nat -A SHADOWSOCKS -d 1.0.9.7 -j RETURN
#iptables -t nat -A SHADOWSOCKS -d 1.0.9.6 -j RETURN

# Ignore LANs IP address
iptables -t nat -A SHADOWSOCKS -d 0.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 10.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 127.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 169.254.0.0/16 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 172.16.0.0/12 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 192.168.0.0/16 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 224.0.0.0/4 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 240.0.0.0/4 -j RETURN

# Check whitelist
iptables -t nat -A SHADOWSOCKS -j SHADOWSOCKS_WHITELIST
iptables -t nat -A SHADOWSOCKS -m mark --mark 1 -j RETURN

#for hulu.com
```

```
iptables -t nat -A SHADOWSOCKS -p tcp --dport 1935 -j REDIRECT --to-ports 7654
iptables -t nat -A SHADOWSOCKS -p udp --dport 1935 -j REDIRECT --to-ports 7654

# for Chrome browser and youtube.com
iptables -t nat -A SHADOWSOCKS -p udp --dport 443 -j REDIRECT --to-ports 7654

# Anything else should be redirected to shadowsocks's local port
iptables -t nat -A SHADOWSOCKS -p tcp -j REDIRECT --to-ports 7654
# Apply the rules
iptables -t nat -A PREROUTING -p tcp -j SHADOWSOCKS

# Ignore China IP address
for white_ip in `cat /etc/chinadns_chnroute.txt`;
do
    iptables -t nat -A SHADOWSOCKS_WHITELIST -d "${white_ip}" -j MARK --set-mark 1
done

# Ignore Asia IP address
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 1.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 14.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 27.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 36.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 39.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 42.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 49.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 58.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 59.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 60.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 61.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 101.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 103.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 106.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 110.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 111.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 112.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 113.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 114.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 115.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 116.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 117.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 118.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 119.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 120.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 121.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 122.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 123.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 124.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 125.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 126.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 169.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 175.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 180.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 182.0.0.0/8 -j MARK --set-mark 1
```

```
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 183.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 202.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 203.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 210.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 211.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 218.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 219.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 220.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 221.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 222.0.0.0/8 -j MARK --set-mark 1
#iptables -t nat -A SHADOWSOCKS_WHITELIST -d 223.0.0.0/8 -j MARK --set-mark 1
```

OpenWrt路由器 iptables防火墙设置含义

- 如果本地发出请求到shadowsocks服务端所在的服务器,就返回，不作任何特殊处理。
- 如果本地发出请求到局域网，也立即返回
- 如果发出请求到中国的IP地址，也立即返回
  chinadns_chnroute.txt是中国IP地址，见 https://github.com/softwaredownload/openwrt-fanqiang/blob/master/openwrt/default/etc/chinadns_chnroute.txt
  预编译翻墙固件都带了这个文件。这个文件很长，因此配置不高的路由器DIR-505，预编译固件里改成了“发出请求到亚洲的IP地址就立即返回”，见文件
  https://github.com/softwaredownload/openwrt-fanqiang/blob/master/openwrt/dir505/usr/bin/shadowsocks-firewall
- 剩下的IP内容请求，全部转发到shdowsocks-libev本地客户端ss-redir监听的端口，由ss-redir负责和服务端进行加密通讯。（手下报告访问youtube的屁民为个位数，领导心里那个高兴啊。可惜经过加密，内容传输速度会有下降）
- 首先运行全代理模式，然后再执行白名单。在白名单比较长时冷启动的速度会比较快。（Thanks Phoeagon)
- 中国的IP列表比较长，如果你的路由器硬件配置不是太好，可以把Ignore China IP address段注释掉，启用Ignore Asia IP address段
- 据tefiszx建议，upd协议 443端口流量转发到shadowsocks的本地端口，请大家测试并反馈(20170711)

OpenWrt路由器防火墙设置重要说明：

- 你必须把上面的1.0.9.8换成你服务器真实的IP地址
- *iptables -t nat -A SHADOWSOCKS -p tcp -j REDIRECT --to-ports 7654* 这里的7654必须和OpenWrt路由器 /etc/shadowsocks.json里的 *local_port*一样，也就是说，如果 /etc/shadowsocks.json里 *"local_port":1090,* 那这里的7654也要改成1090
- 其他可以保持默认

控制shadowsocks本地客户端的方法

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/shadowsocks enable
root@OpenWrt:~# /etc/init.d/shadowsocks disable
```

说明：

- stop: 停止shdowsocks
- start: 运行shadowsocks
- enable: 设置shadowsocks在OpenWrt路由器启动时自动启动
- disable: 取消shadowsocks随机启动

启动并测试shadowsocks-libev本地客户端

确保所有设置无误后，可以启动测试一下：

```
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/shadowsocks enable
```

然后在Ubuntu电脑，手机等设备上打开youtube.com，twitter.com

下载配置文件的最新版

```
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

git clone 项目到本地后，可以进入 openwrt目录查看文件。

如果所有设置都正确，应该可以较快速度打开被墙网站。

OpenWrt自动更新设置和屏蔽广告

OpenWrt路由器自动更新国内重要网站名单

登录路由器后：

```
root@OpenWrt:~# cd /usr/bin
root@OpenWrt:~# touch chinalist
root@OpenWrt:~# chmod +x chinalist
root@OpenWrt:~# vi chinalist
```

/usr/bin/chinalist：

```
#!/bin/sh


wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/accelerated-domains.china.conf
wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/bogus-nxdomain.china.conf
```

OpenWrt路由器自动屏蔽广告

/etc/dnsmasq.d下有个 blockad.conf 文件，内容类似如下：

```
server=/.mobads.baidu.com/127.0.0.0
server=/.mobads-logs.baidu.com/127.0.0.0
server=/.media.admob.com/127.0.0.0
...
```

意思是.mobads.baidu.com的域名解析转发到 127.0.0.0，这个地址不具备域名解析的功能，于是就达到了屏蔽广告的功能。

运行命令：

```
root@OpenWrt:~# cd /usr/bin
root@OpenWrt:~# touch blockad
root@OpenWrt:~# chmod +x blockad
root@OpenWrt:~# vi blockad
```

/usr/bin/blockad：

```
#!/bin/sh

# Author:    https://github.com/softwaredownload/openwrt-fanqiang
# Date:        2016-01-09

TMP_HOSTS=/tmp/block.hosts.unsorted
HOSTS=/etc/dnsmasq.d/blockad.conf

# remove any old TMP_HOSTS that might have stuck around
rm ${TMP_HOSTS} 2> /dev/null

for URL in \
    "https://raw.githubusercontent.com/vokins/simpleu/master/hosts" \
    "http://adaway.org/hosts.txt"
do
    # filter out comment lines, empty lines, localhost...
    # remove trailing comments, space( ,tab), empty line
    # replace line to dnsmasq format
    # remove carriage returns
    # append the results to TMP_HOSTS
    wget -4 --no-check-certificate -qO- "${URL}" | grep -v -e "^#" -e "^\s*$" -e "loca
lhost" -e "^;" -e "^@" -e "^:" -e "^[a-zA-Z]" \
    | sed -E -e "s/#.*$//" -e "s/[[:space:]]*//g" -e "/^$/d" \
    -e "s/^127.0.0.1/server=\/./" -e "s/0.0.0.0/server=\/./" -e "/^[0-9].*$/d" -e "s/$
/\/127.0.0.0/" \
    | tr -d "\r" >> ${TMP_HOSTS}

done

# remove duplicate hosts and save the real hosts file
sort ${TMP_HOSTS} | uniq > ${HOSTS}

rm ${TMP_HOSTS} 2> /dev/null
```

OpenWrt自动生成广告屏蔽列表说明：

- 第一个URL主要用于国内，下面几个URL是屏蔽国外广告
- 运行上面命令产生的广告屏蔽列表比较长，如果路由器性能比较低，dnsmasq匹配域名负荷会太大，可以用直接用下面这个简化版的文件，不要用上面的脚本：
  https://github.com/softwaredownload/openwrt-fanqiang/blob/master/openwrt/dir505/etc/dnsmasq.d/blockad.conf
- 如果dnsmasq超负荷工作，可能会失去响应，导致打不开网页，这时需要登录路由器运行命令：
  /etc/init.d/dnsmasq restart
- 所以，还是尽量用性能好点的路由器吧

路由器性能比电脑差很多，如果屏蔽列表很长，那么短时间内快速打开数个网页就可能导致dnsmasq失去响应。最好是看完一个网页就关闭一个，再打开新的网页。

在路由器里屏蔽的好处是所有接入路由器的设备都全部起作用。

通常的做法，在路由器里屏蔽部分域名，然后在电脑里设置更广泛、精确的屏蔽，主要是设置host文件屏蔽和浏览器插件屏蔽。

浏览器插件屏蔽，可以装这些Chrome浏览器插件：uBlock Origin，ADfree.Player.Online。其中uBlock Origin的作用和Adblock Plus类似，但是设置更加丰富。

计划任务：定时更新dnsmasq配置文件和自动重启shadowsocks

```
root@OpenWrt:~# crontab -e
```

输入以下内容：

```
*/30 * * * * isfound=$(ps | grep "ss-redir" | grep -v "grep"); if [ -z "$isfound" ]; then echo "$(date): restart ss-redir...">>/tmp/log/ss-monitor.log && /etc/init.d/shadowsocks restart; fi
* 12 * * * /usr/bin/chinalist
* 12 * * * /usr/bin/blockad
```

OpenWrt计划任务说明：

- 每半小时检查shadowsocks-libev 客户端，如果退出就自动重启
- 每天中午12点运行chinalist
- 每天中午12点运行blockad

2014-09-24版的dir505, wr2543预编译固件是启用了计划任务的，这会有潜在的不确定性，如果更新时下载的文件如accelerated-domains.china.conf存在错误，导致dnsmasq无法启动，翻墙功能自然失效。

如果你启用了计划任务，某一天突然不能翻墙了，这时设置客户端的IP地址为和路由器同网段，登录路由器，用ps命令查看dnsmasq进程是否启动了，如果没有启动，就重刷固件或者用
https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/dnsmasq.d
下面的文件代替 路由器里/etc/dnsmasq.d/下的文件。

附录：计划任务定时关闭路由器OpenWrt

人类的本性是目光短浅，玩得一时兴趣就会忘记定时休息的重要性。解决办法是在路由器里设置计划任务，禁止夜里某个时间段里使用路由器。下面的例子中，每20分钟检测一次，如果迟于20点30或者早于7点就自动关闭OpenWrt。这对小孩子特别有用，现在很多孩子使用电子设备上瘾，一个人睡的话甚至半夜在被窝里偷偷上网，现在好了，除非孩子强大到会登陆路由器修改设置，否则半夜重启路由器都无法上网了。

```
*/20 * * * * TIME=$(date +"%H%M"); if [ $TIME -ge 2030 ] || [ $TIME -le 700 ]; then po
weroff; fi
```

参考：

- https://github.com/vokins/simpleu
- https://github.com/jjack/openwrt-adblock
- https://github.com/felixonmars/dnsmasq-china-list
- install-shadowsocks-on-hg255d-openwrt-and-config-nat

OpenWrt路由器为什么会翻墙失败或不稳定

给路由器刷上OpenWrt，并按照 本教程设置了服务端和客户端，但还是不能翻墙，或者不稳定，有时能翻，有时不能翻，怎么办？

ping 服务器的ip 看看速度怎么样

```
ping 1.0.9.8
```

检查shadowsocks服务端启动时有没有带上 -u参数

> -u enable udprelay mode
> TPROXY is required in redir mode

本教程使用的，也就是官方的shadowsocs-libev服务端是默认启动带上 -u 参数的。但有的朋友可能使用其他版本的服务端，如Python版，就不能保证服务端启动时默认就带 -u 参数。

可以这样查询服务端是否启动，及启动参数：

```
$ ps -aux | grep ss-server
#.../usr/bin/ss-server -c /etc/shadowsocks-libev/config.json -a root -u -f /var/run/sh
adowsocks-libev/shadowsocks-libev.pid
```

可见上面启动时已经带了 -u 参数。

登录OpenWrt路由器查询翻墙相关进程有没有启动

```
root@eastking:~# ps | grep ss-
#.../usr/bin/ss-redir -b 0.0.0.0 -c /etc/shadowsocks.json -f /var/run/shadowsocks.pid
#.../usr/bin/ss-tunnel -b 0.0.0.0 -c /etc/shadowsocks.json -l 3210 -L 8.8.8.8:53 -u

root@eastking:~# ps | grep dnsmasq
#.../usr/sbin/dnsmasq -C /var/etc/dnsmasq.conf -k -x /var/run/dnsmasq/dnsmasq.pid
```

上面的查询显示，ss-redir ss-tunnel dnsmasq都已经正常启动。

有时虽然ss-redir ss-tunnel dnsmasq等进程都在，但已经失去响应了，这就需要：

重启 shadowsocks，登录路由器，运行命令：

```
/etc/init.d/shadowsocks restart
```

`restart` 内部分 `stop` 和 `start` 两步执行，实际测试发现，少数时候 `stop` 并不能关闭shadowsocks相关进程，那么只能：

按路由器背后的电源重启OpenWrt路由器

翻墙不稳定，有时能连上被墙网站，有时连不上

shadowsocks-libev加密翻墙的方式加大了墙的辨识难度，但不是不可能被辨识。因此，还是有可能受到干扰的。解决方法：更换加密方式，如改成 `bf-cfb`

一般情况下这样就能解决问题。

登录路由器用dig查询被墙域名

本教程预编译的翻墙固件都安装了 bind-dig，方便调试。

注：本教程默认的 tunnel 转发端口都是 3210

正常的结果类似如下：

```
root@eastking:~# dig @localhost -p 3210 google.com

; <<>> DiG 9.9.7-P3 <<>> @localhost -p 3210 google.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38460
;; flags: qr rd ra; QUERY: 1, ANSWER: 11, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             299     IN      A       74.125.226.33
google.com.             299     IN      A       74.125.226.36
google.com.             299     IN      A       74.125.226.32
google.com.             299     IN      A       74.125.226.38
google.com.             299     IN      A       74.125.226.41
google.com.             299     IN      A       74.125.226.39
google.com.             299     IN      A       74.125.226.35
google.com.             299     IN      A       74.125.226.46
google.com.             299     IN      A       74.125.226.37
google.com.             299     IN      A       74.125.226.40
google.com.             299     IN      A       74.125.226.34

;; Query time: 290 msec
;; SERVER: 127.0.0.1#3210(127.0.0.1)
;; WHEN: Mon Dec 28 11:55:30 CST 2015
;; MSG SIZE  rcvd: 215
```

更换服务器IP地址

有时 ssh 能连上服务器，登录路由器用dig查询被墙域名没有结果。出现这种情况有可能是服务器IP被限制了，更换IP就可能恢复正常

仔细检查每一项翻墙配置

还是不能翻墙？也有可能是某项配置有误，仔细检查教程中讲到的每一项翻墙设置，确保没有错误

有一次我给路由器新刷翻墙固件后，总是不能翻墙。于是逐项检查，发现了某项配置有误，修正后就可用了

延伸阅读：

ShadowSocks教程:shadowsocks是否支持udp转发是什么来的

udp是什么：UDP 是User Datagram Protocol的简称， 中文名是用户数据报协议，是OSI（Open System Interconnection，开放式系统互联） 参考模型中一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务，IETF RFC 768是UDP的正式规范。UDP在IP报文的协议号是17。 UDP协议全称是用户数据报协议[1] ，在网络中它与TCP协议一样用于处理数据包，是一种无连接的协议。在OSI模型中，在第四层——传输层，处于IP协议的上一层。UDP有不提供数据包分组、组装和不能对数据包进行排序的缺点，也就是说，当报文发送之后，是无法得知其是否安全完整到达的。UDP用来支持那些需要在计算机之间传输数据的网络应用。包括网络视频会议系统在内的众多的客户/服务器模式的网络应用都需要使用UDP协议。UDP协议从问世至今已经被使用了很多年，虽然其最初的光彩已经被一些类似协议所掩盖，但是即使是在今天UDP仍然不失为一项非常实用和可行的网络传输层协议。 与所熟知的TCP（传输控制协议）协议一样，UDP协议直接位于IP（网际协议）协议的顶层。根据OSI（开放系统互连）参考模型，UDP和TCP都属于传输层协议。UDP协议的主要作用是将网络数据流量压缩成数据包的形式。一个典型的数据包就是一个二进制数据的传输单位。每一个数据包的前8个字节用来包含报头信息，剩余字节则用来包含具体的传输数据。

shadowsocks-android 的 DNS (UDP) 转发功能

从 2.1.2 开始，shadowsocks-android 开始支持透明的 DNS (UDP) 转发功能。这项功能包括两个部分：

1. NAT（ROOT）模式下，仅支持转发 DNS 的 UDP 数据包。
2. VPN 模式下，支持转发所有的 UDP 数据包。

限制：

1. 当前只有 1.4 以上的 libev 或 nodejs 实现的服务器端才支持此项功能。
2. libev 服务器端还需要在命令行中加上 -u 的参数。
3. 此项功能默认关闭，依然由 pdnsd 负责转发 TCP 的 DNS 查询。

网友提出的几个问题：

问题一：shadowsocks-libev 默认启用了 udp relay 吗？

请问udp relay功能是否有必要打开？我看shadowsocks-android是有这个选项支持该功能的，但shadowsocks-qt5貌似不支持。 另外，config.json里面是不是不支持写明是否需要打开udp relay，而必须要ss-server -c /etc/shadowsocks/config.json -u这么写吗？

debian下文件在/etc/init.d/shadowsocks-libev，找到

```
start-stop-daemon –start –quiet –pidfile $PIDFILE –chuid $USER:$GROUP –exec $DAEMON –
\
-c “$CONFFILE” -a “$USER” -u -f $PIDFILE $DAEMON_ARGS \
```

发现已经默认加上-u参数，1.6.1版测试结果

问题二：shadowsocks android vpn 模式要避免 dns 污染要打开 UDP 转发吗？

国内ps4联机很蛋疼，于是在路由器里搞了一个支持shadowsocks的固件，作者说支持udprelay，会转发udp数据包。 于是我就在我的服务器里开通了一个支持udprelay的ss账号，用的是最新版的ss-libev，启动参数中加了-u，应该没错。 实测ps4也可以打开youtube，但是ps4网络测试结果为nat类型失败。 我不确定是路由器固件作者的问题还是ss-libev的udprelay功能有bug,所以我需要一个可以很好支持udprelay的ss账号，进行测试。

问题三：shadowsocks 现在能不能代理游戏，我看说支持 UDP 了？

对代理游戏有一定需求（MAC版的美服 BATTLE.NET），现在SS能不能直接全局代理游戏，搜索了下貌似之前的一个版本就添加了对UDP的支持且默认开启，是不是意思是开了全局模式就默认代理UDP/TCP了？

问题四：不确定 SS 服务器端是否支持 UDP 转发，有办法测试么？

买了个套装服务，内含SS，找了个703N的路由器刷了openwrt官方镜像开始一步一步安装shadowsocks-libev版，我的想法是用这个703N做全局翻，所以DNS解析也用udp转发到8.8.4.4:53，但测了半天不好用，才想起对面的SS服务器端未必开了这个功能，现在我如何确定服务器端是否打开了UDP转发？ 或者UDP转发这个功能压根和服务器端没关系？

```
ss-tunnel.exe -c config.json -l 53 -L 8.8.8.8:53 -u
nslookup www.youtube.com 127.0.0.1
```

如有返回结果则开启了udp转发

/etc/init.d/shadowsocks 这个脚本里本身已经设置了 -u ，不是这样执行的。如果你要手动加 -u, 则是 ss-server -c /etc/shadowsocks/config.json -u

Shadowsocks翻墙不同加密方式速度区别

Shadowsocks翻墙不同加密方法，哪一种速度最快最好：

- 翻墙不稳定，有的能上，有的不能上，有时能上，有时不能上，可能是加密方式的特征被识别，从而被干扰，方法是更换加密方式
- rc4-md5加解密速度虽然快，但是加密强度不够大，容易被干扰
- 无论哪一种加密方式，只要使用的人多了，就可能被重点研究，从而受到干扰
- aes-256-cfb 加密强度大些，一样可能被干扰
- 有人推荐chacha20或者salsa20，没有试过
- 其实 bf-cfb就很好，速度很快，官方的shadowsocks-libev及本教程预编译的翻墙固件都直接支持

网友实测Shadowsocks不同加密方尘速度

encrypt and decrypt 20MB data:

```
aes-128-cfb 0.368462085724s
aes-128-ofb 0.400309085846s
aes-192-cfb 0.452577829361s
aes-192-ofb 0.381041049957s
aes-256-cfb 0.418514966965s
aes-256-ofb 0.405379056931s
cast5-cfb 0.859935045242s
cast5-ofb 0.911785125732s
chacha20 0.429271936417s
rc4 0.154517173767s
rc4-md5 0.169504165649s
salsa20 0.44139790535s
```

网友评论：

clowwindy：

> 因为 chacha20 从 x86 上的性能来看，对速度的影响太小，提高还太有限，不如换个思路，因为通信包到了终端以后，走的都是电路，这里其实涉及到一个供电体系的问题，更换加密不如换一个电网，同一个 VPS，同一个路由器，但是，改用核电给路由器供电时，比火电丢包率会降低一个数量级，大大提高 TCP 吞吐率，经测试这是目前速度最快的供电方式，甚至优于水电，同理选 VPS 机房也要看供电，有些号称用了绿色能源，其实效果不好，这里面其实还涉及到选用 UPS 的型号，就不细说了 另外说到硬件加速，连接路由器的网线也很重要，建议用六类屏蔽线，不过一定不能买那种超薄扁平的网线，会对带宽起到整形作用，突发上不去，看 4K 会受影响，数据可能不准，不过大概也体现了差异 今天没有时间再次测试了，就发这么多吧。你们有一个好，出个新功能，写教程比别人都快，但试来试去的结论，太简单，有时太朴素了，我感觉你们还需要学习，提高自己的知识水平，将来如果写的教程有偏差，你们要负责

rlei:

> 讨论shadowsocks不同加密方式的安全性没有意义。shadowsocks是被设计来混淆数据，增加某Wall检查出流量特征所需的计算量，提高实时检测的成本，而不是加密。ss的作者多次强调过这一点

参考：

- https://www.zhihu.com/question/28252105

零起点DO VPS shadowsocks-libev 翻墙设置教程

Digital Ocean 的优点：

- 业界最有名的VPS服务商，服务有保障
- 全SSD硬盘，速度极快，重启在20秒内
- 所有VPS具有独立IP
- 费用极低，$5/月起
- 管理后台Console Access可以直接运行所有linux命令，可以不设置SSH
- 收费以小时计算，不用了可以删除，不会多收一分钱
- 更换IP方便，创建snapshot，再从snapshot新建Droplet,就可能得到新的IP了

立即点击这里注册DO

创建翻墙用的虚拟服务器Droplet

注册DO并绑定支付方式后，登录管理后台，点击右上角的 `Create` 从下拉菜单中选择 `Droplet`：

- Choose an image 选择最新版的Ubuntu 64位，下图是14.04.3，下拉还有更新的如17.04：

Create Droplets

Choose an image ?

Distributions One-click Apps Snapshots

Ubuntu
14.04.3 x64

FreeBSD
Select Version

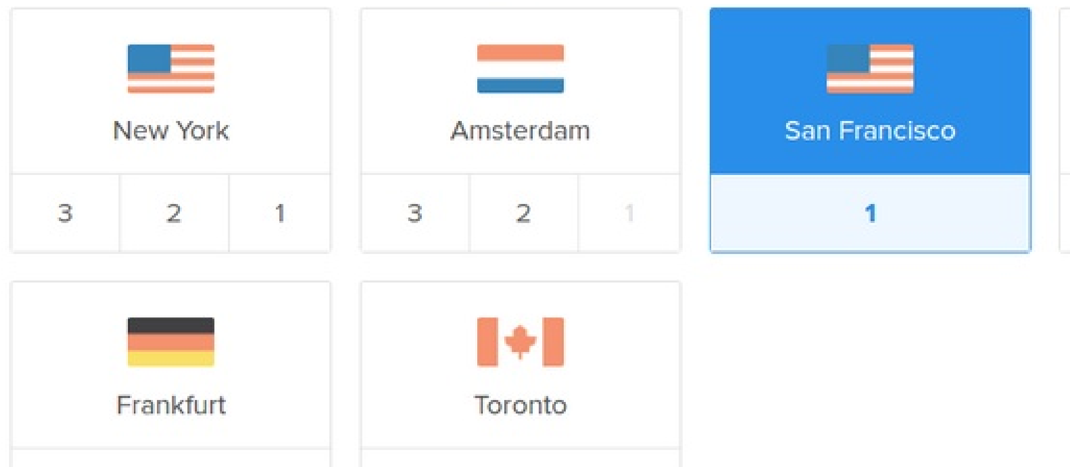
- Choose a site 一般512MB那款就够了：

Choose a size

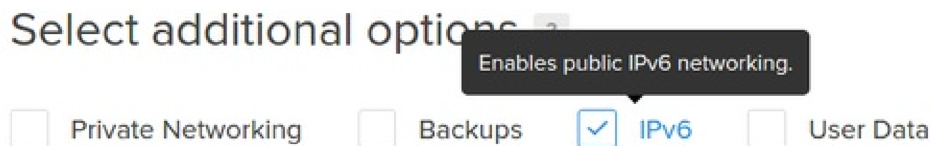
$5/mo
$0.007/hour
512 MB / 1 CPU
20 GB SSD Disk
1000 GB Transfer

$10/mo
$0.015/hour
1 GB / 1 CPU
30 GB SSD Disk
2 TB Transfer

- Choose a datacenter region 经测试选择New York时访问YouTube较快：



- Select additional options, 勾选IPv6：



- Choose a hostname, 只是助记，比如改成ubuntu-shadowsocks
- Create 创建虚拟服务器

进入DO VPS管理界面

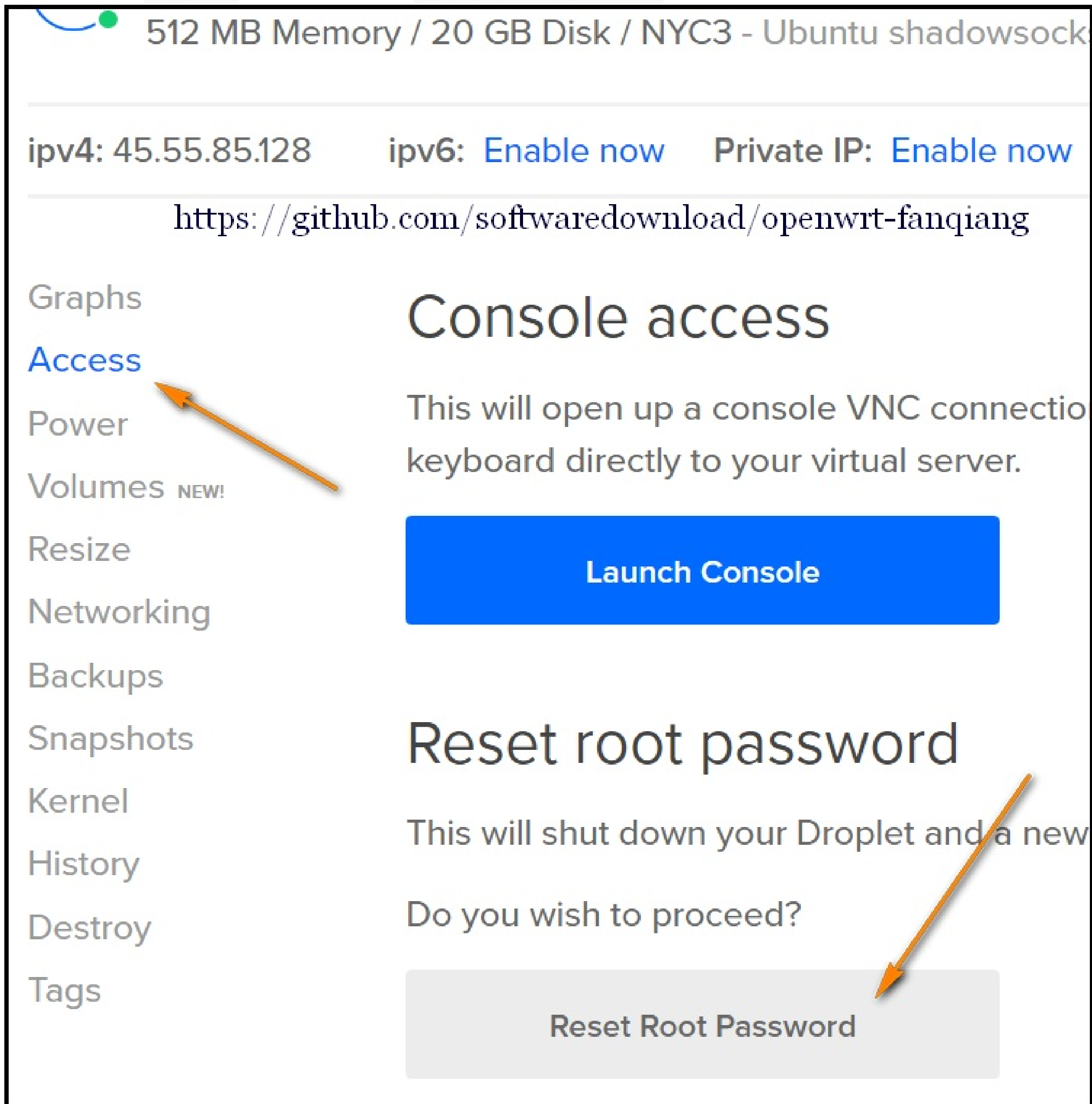
在20秒内，VPS创建完毕后自动进入了 `Droplets` （VPS）列表页面，点击VPS名字进入VPS管理界面：



如何重置DO VPS Root密码:

注：如果已经通过邮件收到root密码，请跳到一下步

点击左边的 `Access` 再点击右边的 `Reset Root Password` 重置密码：

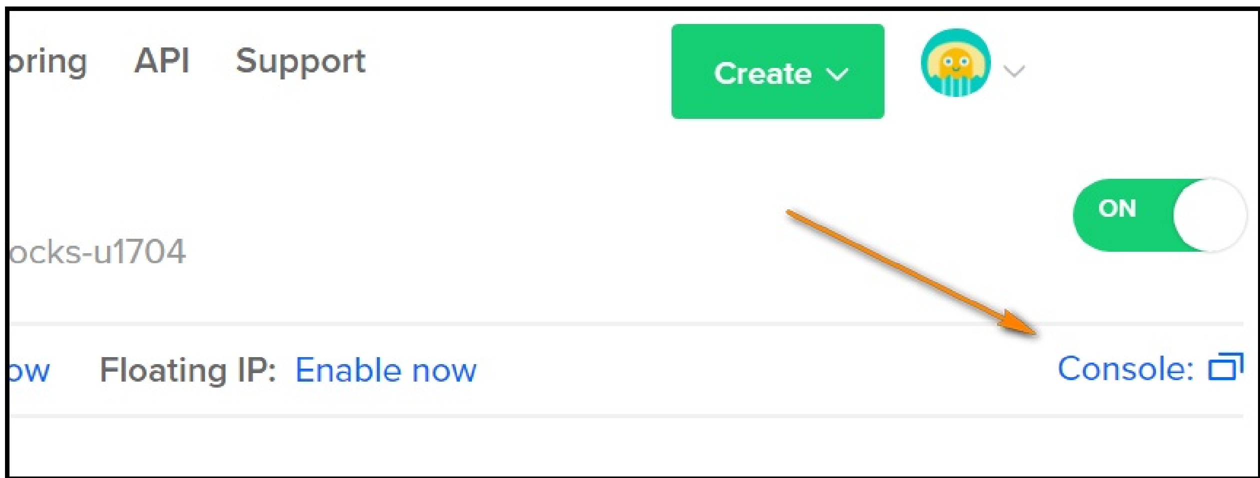


重置密码完成后，新的密码会发送到你的邮箱，下面我们就用这个密码登录并直接通过网页Console 管理VPS

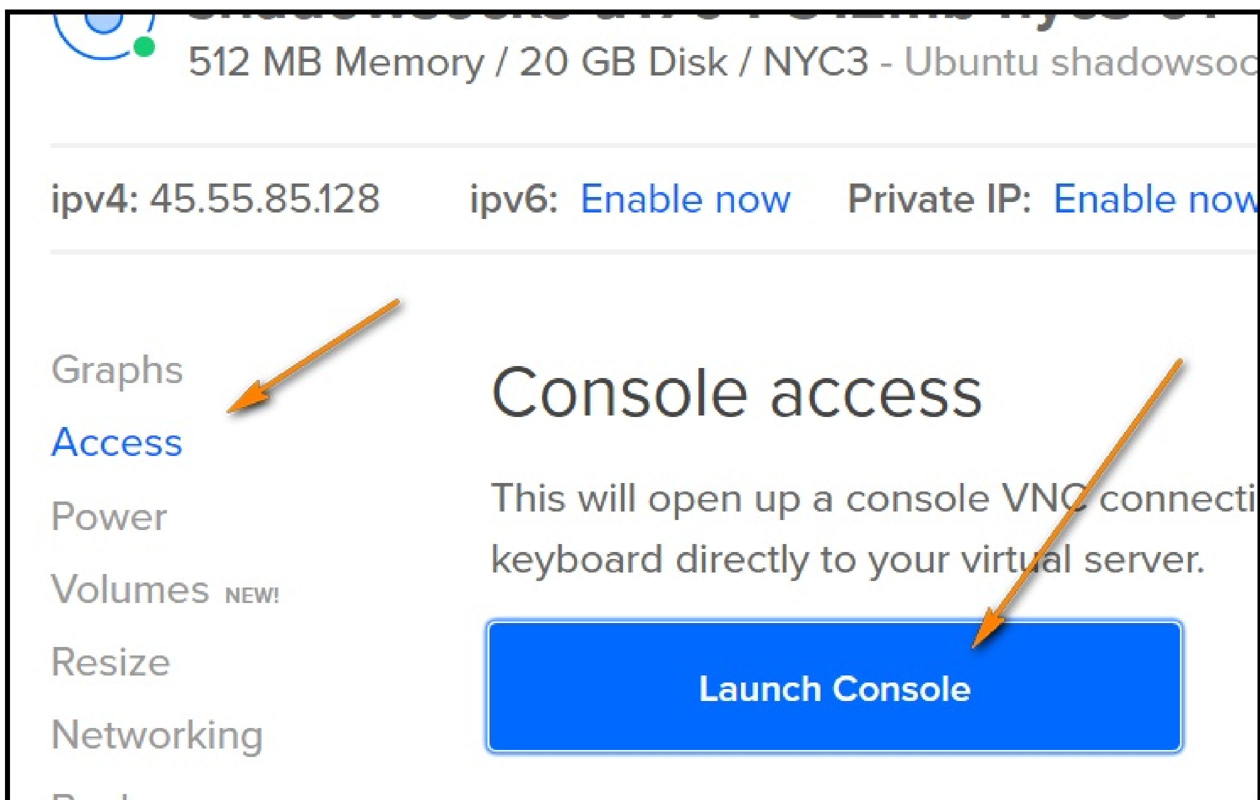
进入DO VPS命令行控制界面 Console Access

DO有个极为强大的功能，可以可以直接在管理后台Console Access 运行Linux命令管理VPS，相当于一个在线版的ssh

点击页面右上角的 `Console` 可以直接进入，如下图:



打开 Console 的另一方法是先点击左边的 `Access` 然后点击 `Launch Console` 如下图：



如果等了一会儿命令行界面还没有出来，就按F5刷新页面直到打开。

点击打开的命令行窗口以获得输入焦点。

命令行设置新的Root密码

开启DO Console Access后，输入root并回车，然后重新设置密码。

```
Ubuntu 17.04 ubuntu-shadowsocks tty1
ubuntu-shadowsocks login: root
Passwd: 输入root密码
You are required to change your password immediately (root enforced)
Changing password for root.
(Current) UNIX password: 输入root密码
Enter new UNIX password: 输入新的root密码
Retype UNIX password: 再次输入新的root密码
```

密码更新完成后更新一下系统：

```
root@ubuntu-shadowsocks:~# apt-get update
root@ubuntu-shadowsocks:~# apt-get dist-upgrade
```

可能会问你要不要更新一下grub，直接回车就行了。（我选择的是升级到 `install the package maintainer's version` ）

从源码编译 shadowsocks-libev server

2016-01-19发现，shadowsocks.org网页无法打开，这给 `apt-get install` 方式安装shadowsocks-libev带来不便，不过我们可以自己从源码编译，很简单，而且随时可以编译到最新的版本。

Console Access界面是无法粘贴命令的，把下面命令逐行粘贴到浏览器地址栏，抄着输入也是很快的，输入第一行命令并回车后输入 y 安装所有相关包。

```
root@ubuntu-shadowsocks:~# apt-get install build-essential autoconf libtool libssl-dev
 gawk debhelper dh-systemd init-system-helpers pkg-config git
root@ubuntu-shadowsocks:~# git clone https://github.com/shadowsocks/shadowsocks-libev.
git
root@ubuntu-shadowsocks:~# cd shadowsocks-libev
root@ubuntu-shadowsocks:~# dpkg-buildpackage -us -uc -i
root@ubuntu-shadowsocks:~# cd ..
root@ubuntu-shadowsocks:~# sudo dpkg -i shadowsocks-libev*.deb
root@ubuntu-shadowsocks:~# ls /usr/bin/ss-*
root@ubuntu-shadowsocks:~# ss-local ss-manager ss-redir ss-server ss-tunnel
```

设置 shadowsocks-libev server，见 翻墙软件Shadowsocks-libev服务端设置

至此，我们已经开通了DO VPS,并且在网页界面就安装完成了 shadowsocks-libev，下面是修改设置并重启shadowsocks-libev

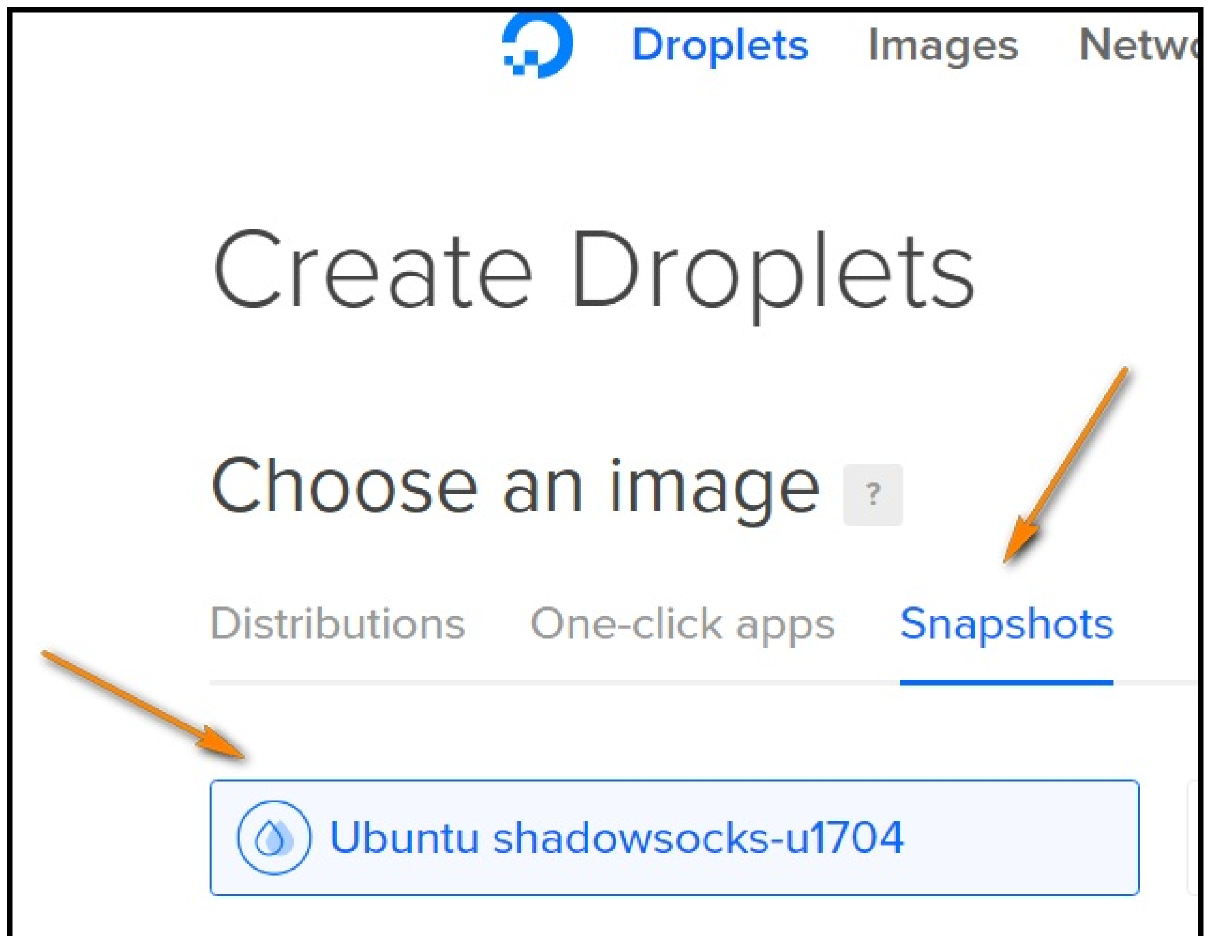
```
root@ubuntu-shadowsocks:~# vi /etc/shadowsocks-libev/config.json
root@ubuntu-shadowsocks:~# service shadowsocks-libev restart
```

详细的设置教程在 翻墙软件Shadowsocks-libev服务端设置

再配置好客户端，如果没有错误，就可以成功翻墙了。所有以上过程2016-01-19亲测通过。

附录：怎样更换DO翻墙VPS的IP（或者怎样使用最省钱）

- 照上面教程创建Droplet ubuntu-shadowsocks, 设置好shadowsocks-libev服务端，其中server写 `0.0.0.0` 并测试通过
- Poweroff VPS，也就是VPS关机，这时还会产生VPS使用费用的，因为IP，空间等资源还是被你占用
- 创建Snapshot，命名为shadowsocks，并传送到你可能使用的各个区域。比如你原来是在San Francisco创建的，可以传送到New York区
- 删除VPS：Destroy Droplet ubuntu-shadowsocks，然后就不产生任何费用了。不怕麻烦，每天都这样操作，一个月可能只要2元钱就行了
- 下次要使用，在Create Droplet的第一步，Choose an image, 选择Snapshots, shadowsocks，其他和上面教程一样。见下图：



- 从snapshot创建Droplet完成，页面显示了VPS的IP地址，shadowsocks客户端连接到这个IP地址就行了，服务端不用更改任何设置

附录：怎样不“登录”路由器更改OpenWrt shadowsocks-libev路由器的server IP

- 路由器设置密钥登录，这样ssh登录就不用密码了
- 创建config配置文件, Ubuntu下是 `~/.ssh/config` ，增加如下内容：

  ```
  Host router
      HostName 192.168.1.1
      User root
      Port 22
      IdentityFile /path/to/your/rsa
  ```

  Windows下安装 git for Windows，选择使用OpenSSH，编辑 `C:\Program Files\Git\etc\ssh\ssh_config`

  然后就可以 `ssh router` 登录路由器了

- reset.sh:

  ```
  #!/bin/bash

  # Author: https://github.com/softwaredownload/openwrt-fanqiang
  # Date: 2016-01-20

  ssh router <<'ENDSSH'

  sed -ri "s/([0-9]{1,3}\.){3}[0-9]{1,3} -j/1.0.9.8 -j/" /usr/bin/shadowsocks-firewall
  sed -ri "s/([0-9]{1,3}\.){3}[0-9]{1,3}/1.0.9.8/" /etc/shadowsocks.json

  /etc/init.d/shadowsocks restart

  ENDSSH
  ```

  把reset.sh中的 `1.0.9.8` 改成shadowsocks服务端的server IP，然后运行 reset.sh就可以了。

  想要测试一下日本，英国，新加坡或美国的IP，so easy，2分钟就行了。

Reference:

- https://github.com/shadowsocks/shadowsocks-libev

OpenWrt编译翻墙固件教程

实践前面的教程，翻墙已经不是问题，白脸也很happy。在这一章中，我们要定制自己OpenWrt固件，刷上定制的固件，不用任何设置就自动翻墙并自动更新规则。

最简单的路由器刷OpenWrt固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读OpenWrt翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

编译shadowsocks-libev for OpenWrt ipk安装包

不同OpenWrt版本下编译的shadowsocks-libev ipk一般是不能通用的。比如现在用的是trunk版的OpenWrt，如果使用OpenWrt Chaos Calmer 15.05 下编译的shadowsocks-libev，可能安装后根本不能启动。

前面我曾编译出翻墙固件，其中shadowsocks-libev是别人编译，从sourceforge上下载的，刷上固件后，shadowsocks总是没有自动启动，运行/usr/bin/ss-redir，报告没有找到这个文件，其实文件是在的，只是不兼容。所以，最好还是自行编译shadowsocks-libev。

以下 不要使用**root**用户来操作

下面是在Ubuntu 64bit下编译shadowsocks-libev for OpenWrt ipk安装包的步骤：

安装依赖库，不同的操作系统版本可能要作相应调整

```
sudo apt-get install build-essential subversion libncurses5-dev zlib1g-dev gawk gcc-mu
ltilib flex git-core gettext
```

下载OpenWrt源代码

```
cd ~/Downloads
git clone git://git.openwrt.org/openwrt.git
```

下载shadowsocks-libev源码

```
cd ~/Downloads/openwrt
pushd package
git clone https://github.com/shadowsocks/shadowsocks-libev.git
popd
```

或者：

```
cd ~/Downloads/openwrt/package
git clone https://github.com/shadowsocks/shadowsocks-libev.git

编译 DIR505固件2015-12版时用的源码版本是：Date:   Tue Dec 22 21:42:40 2015
```

更新Feeds，使package在make menuconfig中可用，而不是眞正安装或编译，并按照自己的路由型号设定target，否则默认target下编译好的工具链在重新设定target后无效

```
cd ~/Downloads/openwrt
./scripts/feeds update -a
./scripts/feeds install -a
# run make menuconfig and set target;
# Choose your own Target System -> SubTarget -> Target Profile
make menuconfig
make defconfig
```

先编译要用到的工具和库

```
make prereq && make tools/install && make toolchain/install
```

等待时间较长，可以先和大妈一起去跳个广场舞，制造更多噪音恶心一下别人:)

make menuconfig配置选项

```
# 运行命令
make menuconfig
```

有三个选项：

- y: 编译进固件
- m: 编译出安装包，但不打包进固件
- n: 排除

输入命令 `make menuconfig` 进入配置程序后：

- Target System:

- Atheros AR7xxx/AR9XXX (Default value，不同的路由器，可能选择不同)
  适合： WNDR4300, DIR505A1, TLWR2543
- Subtarget:
  - Generic device with NAND flash
    适合： WNDR4300
  - Generic
    适合： DIR505A1
- Target Profile: (因我们只是编译包，这步可以不选)
- Network, 选择shadowsocks-libev 和 shadowsocks-libev-polarssl, 按m设置为编译独立ipk安装包
- Save && Exit

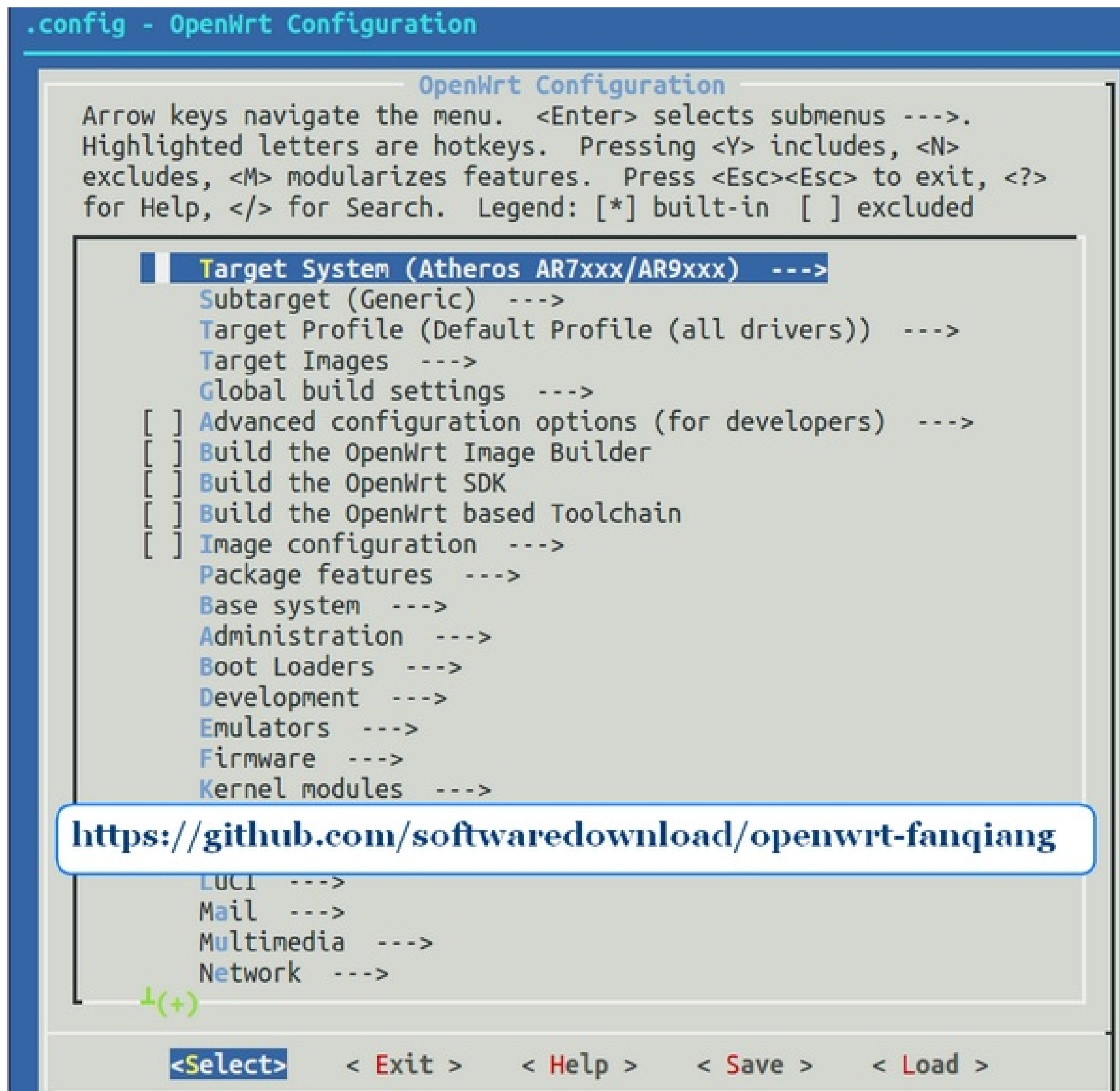


图 make menuconfig

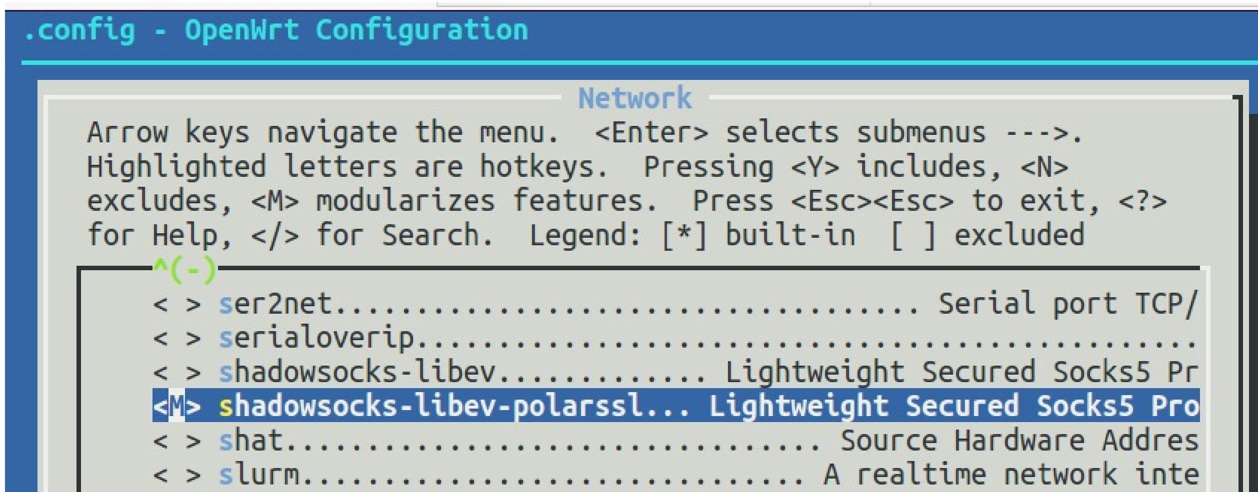


图 选择shadowsocks-libev-polarssl

编译shadowsocks-libev for OpenWrt

```
make V=99 package/shadowsocks-libev/openwrt/compile
```

查看编译出的shadowsocks-libev和shadowsocks-libev-polarssl文件

```
cd ~/Downloads/openwrt/bin/ar71xx/packages/base/
tree
├── libc_1.1.11-1_ar71xx.ipk
├── libgcc_5.2.0-1_ar71xx.ipk
├── libopenssl_1.0.2e-1_ar71xx.ipk
├── libpolarssl_1.3.15-1_ar71xx.ipk
├── libpthread_1.1.11-1_ar71xx.ipk
├── shadowsocks-libev_2.4.3_ar71xx.ipk
├── shadowsocks-libev-polarssl_2.4.3_ar71xx.ipk
└── zlib_1.2.8-1_ar71xx.ipk

# 如果用来编译翻墙固件，把shadowsocks-libev复制到Image Builder目录下：
# for DIR505A1:
cp shadowsocks* ~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64/packages/
base
# for WNDR4300
cp shadowsocks* ~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/packages/bas
e
```

把文件scp复制到OpenWrt路由器/tmp，就可以 `opkg install shadowsocks-libev_2.4.3_ar71xx.ipk` 安装了。

参考：

- http://wiki.openwrt.org/doc/howto/buildroot.exigence
- http://wiki.openwrt.org/doc/howto/build
- https://github.com/shadowsocks/shadowsocks-libev
- http://sourceforge.net/projects/openwrt-dist/files/shadowsocks-libev/
- https://0066.in/archives/312

下载和设置翻墙配置文件

自己手工收集编辑翻墙所用到的配置文件是件比较累的事情。最快的方法是 git clone 本项目，修改其中某些选项。

下载翻墙配置文件

```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

默认配置文件目录：openwrt-fanqiang/openwrt/default

针对特定路由器的配置文件目录，以路由器型号为目录名，如 openwrt-fanqiang/openwrt/wndr4300

复制配置文件，以**wndr4300**路由器为例：

- 本地建立配置文件目录，如 ~/Downloads/openwrt-wndr4300
- 复制默认配置文件到 ~/Downloads/openwrt-wndr4300

  ```
  mkdir ~/Downloads/openwrt-wndr4300

  # Linux下复制默认配置文件
  cp -R ~/Downloads/openwrt-fanqiang/openwrt/default/* ~/Downloads/openwrt-wndr4300/

  # 复制WNDR4300路由器的特定配置文件，同名文件就覆盖
  cp -R ~/Downloads/openwrt-fanqiang/openwrt/wndr4300/* ~/Downloads/openwrt-wndr4300/
  ```

修改配置文件，编译后就直接可以用了。否则刷上固件后登录路由器再修改。主要修改如下文件：

```
~/Downloads/openwrt-wndr4300/etc/shadowsocks.json
~/Downloads/openwrt-wndr4300/usr/bin/shadowsocks-firewall
~/Downloads/openwrt-wndr4300/etc/uci-defaults/defaults
```

- shadowsocks.json 中 server必须改成你的服务器实际IP
- defaults 中wan-username 和 wan-password必改
- shadowsocks-firewall 中 1.0.9.8必须改成你的服务器实际IP

- 编译自定义固件时，设置FILES=~/Downloads/openwrt-wndr4300

自定义配置文件用途说明

定制固件的前提是你要有一台服务器运行shadowsocks服务端ss-server。

- etc/dnsmasq.conf 设置dnsmasq配置文件目录
- etc/shadow 登录路由器的密码， 默认是fanqiang
- etc/uci-defaults/defaults 默认上网设置及时区等设置

关于 /etc/uci-defaults目录

uci-defaults目录下的文件会在路由器第一次启动时由/etc/init.d/boot执行,如果在文件末尾加上exit 0, 则执行就会删除此文件，否则执行成功则删除，不成功则在下次启动时继续执行直到成功。

我们在这个目录下创建一个defaults文件，在这个文件中设置上网参数，时区等。

To set some system defaults the first time the device boots, create a script in the folder

All scripts in that folder are automatically executed by /etc/init.d/boot and if they exited with code 0 deleted afterwards (scripts that did not exit with code 0 are not deleted and will be re-executed during the next boot until they also successfully exit).

默认端口及修改方法（可以不改）：

- shadowsocks服务端监听端口：1098
  - 文件位置： 服务器/etc/shadowsocks-libev/config.json
  - 如更改，路由器里 /etc/shadowsocks.json也相应更改
- 路由器shadowsocks ss-redir 监听端口：7654
  - 文件位置： 路由器/etc/shadowsocks.json
  - 如更改， 路由器/usr/bin/shadowsocks-firewall也相应更改
- 路由器shadowsocks ss-tunnel监听端口: 3210
  - 文件位置: 路由器/etc/init.d/shadowsocks
  - 如更改, 路由器 /etc/dnsmasq.d/gfwlist.conf也相应更改

以上端口建议不改。程序运行稳定后，相关密码可以改掉。

端口关联的理解：

- shadowsocks-firewall负责把非中国流量转发到本地端口7654
- ss-redir监听端口7654，该端口流量都加密走自己的服务器通道
- dnsmasq把非国内重要域名的dns查询转发本地3210端口
- ss-tunnel监听本地端口3210,把该端口的dns查询转发到自己服务器向8.8.8.8查询

设置可执行权限

```
chmod +x usr/bin
chmod +x usr/bin/*
chmod +x etc/uci-defaults
chmod +x etc/uci-defaults/defaults
```

参考：

*http://wiki.openwrt.org/doc/uci

使用Image Builder编译自动翻墙OpenWrt固件

Image Builder又叫Image Generator，利用它我们可以方便地定制适合自己无线路由器的固件。

编译OpenWrt自定义翻墙固件的注意事项

- 不要用“root”用户编译
- 进入到编译系统目录中执行编译相关命令，如：~/Downloads/openwrt
- 在编译版的路径中不能够出现空格
- 如果已经用root用户下载并解压了源码，可用命令改属主成普通用户：sudo chown -R user:user ~/Downloads/openwrt

下载适合自己无线路由器的Image Builder

- 进入 http://downloads.openwrt.org/
- 选择Binary Releases或 Development Snapshots
    - 目前的Binary Releases: http://downloads.openwrt.org/chaos_calmer/15.05/
    - Development Snapshots: http://downloads.openwrt.org/snapshots/trunk/
- 选择 CPU类型，如 ar71xx: http://downloads.openwrt.org/snapshots/trunk/ar71xx/
- 选择 Flash 类型, 如generic: http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/

下载命令举例:

```
cd ~/Downloads
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64.tar.bz2
tar -xjf  OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64.tar.bz2
```

下载包含默认翻墙配置文件的openwrt-fanqiang项目

- git下载openwrt-fanqiang项目

  cd ~/Downloads git clone https://github.com/softwaredownload/openwrt-fanqiang

- 或者下载zip文件

  https://github.com/softwaredownload/openwrt-fanqiang/archive/master.zip

本地项目文件夹是：~/Downloads/openwrt-fanqiang

复制openwrt-fanqiang里面的翻墙配置文件到openwrt-tlwr2543目录下

建立一个配置文件夹，以路由器型号结束，如 ~/Downloads/openwrt-tlwr2543。

```
cd ~/Downloads
mkdir openwrt-tlwr2543

cd openwrt-fanqiang
cp -R openwrt/default/* ~/Downloads/openwrt-tlwr2543/
cp -R openwrt/tlwr2543/* ~/Downloads/openwrt-tlwr2543/
```

上面的操作，先复制共用的配置文件 openwrt/default/ 到 *openwrt-tlwr2543*目录下
然后复制*wr2543*专用的配置文件*(*如果存在*)*到 *openwrt/tlwr2543/* 到 openwrt-tlwr2543目录下，如果有同名文件就覆盖。

如果你要贡献本项目，也是先在openwrt-fanqiang/openwrt目录下先建立路由器型号为名称的文件夹，再把专用的配置文件放到此文夹下。注意文件夹和文件名都是小写的。

修改TL-WR2543路由器翻墙配置文件

主要修改以下文件：

```
openwrt-tlwr2543/etc/shadowsocks.json
openwrt-tlwr2543/usr/bin/shadowsocks-firewall
openwrt-tlwr2543/etc/uci-defaults/defaults
```

为了方便以后升级，可以写个bash文件自动修改配置文件。

一切操作尽量自动化，你甚至可以自动化一切操作：下载ImageBuilder，下载OpenWrt源码，下载shadowsocks-libev源码，同步openwrt-fanqiang源码，编译ipk，修改翻墙设置，编译翻墙固件，早上一觉醒来，新鲜出炉、美味可口的翻墙固件就已经摆放在桌上了。

下面是一个自动修改配置文件的例子，从中可以知道需要修改哪些地方。从2015年12月起，可能用于自动化修改的默认值都应该标准化，方便自动化操作。

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date:   2015-12-24

REPOSITORY=~/Downloads/openwrt-fanqiang
CONFIG=~/Downloads/openwrt-tlwr2543

createdir() {
    rm -rf $CONFIG
    mkdir $CONFIG
}

copy() {
    cp -R $REPOSITORY/openwrt/default/* $CONFIG/
    cp -R $REPOSITORY/openwrt/tlwr2543/* $CONFIG/
}

setmod() {
    chmod +x $CONFIG/usr/bin/shadowsocks-firewall
    chmod +x $CONFIG/etc/uci-defaults
    chmod +x $CONFIG/etc/uci-defaults/*
}

modify() {
    # server ip address
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/etc/shadowsocks.json

    # server_port
    sed -i 's/1098/server_port/' $CONFIG/etc/shadowsocks.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/etc/shadowsocks.json

    # password
    sed -i 's/killgfw/killgfw/' $CONFIG/etc/shadowsocks.json

    # method
    sed -i 's/aes-256-cfb/aes-256-cfb/' $CONFIG/etc/shadowsocks.json


    # server ip addresss
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/usr/bin/shadowsocks-firewall

    # local_port
    sed -i 's/7654/7654/' $CONFIG/usr/bin/shadowsocks-firewall


    # ppoe username
```

```
    sed -i 's/wan-username/wan-username/' $CONFIG/etc/uci-defaults/defaults

    # ppoe password
    sed -i 's/wan-password/wan-password/' $CONFIG/etc/uci-defaults/defaults

    # wifi password
    sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/etc/uci-defaults/defaults

    # root password
    sed -i 's/\\nfanqiang/\\nfanqiang/' $CONFIG/etc/uci-defaults/defaults
}


if [ "$1" = "createdir" ]; then
    createdir
elif [ "$1" = "copy" ]; then
    copy
elif [ "$1" = "setmod" ]; then
    setmod
elif [ "$1" = "modify" ]; then
    modify
else
    echo "usage: createdir copy setmod modify"
fi
```

自动修改翻墙配置文件用法：

```
./config-tlwr2543.sh createdir
./config-tlwr2543.sh copy
./config-tlwr2543.sh setmod
./config-tlwr2543.sh modify
```

确定OpenWrt无线路由器的PROFILE值

```
cd OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64
make info
```

找到自己固件的型号，比如我的是 `TP-LINK TL-WR2543N/ND`,它的PROFILE值是TLWR2543。如下图：

```
TLWR2543:
        TP-LINK TL-WR2543N/ND
        Packages: kmod-usb-core kmod-usb2 kmod-ledtrig-usbdev
```

找出默认应该包含进OpenWrt固件的包

对于TP-LINK WR2543无线路由器来说，可以这样获取：

```
echo $(wget -qO - http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/config |
 sed -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1/ip')
```

由于 OpenWrt开发非常活跃，不同版本的基础包可能是不同的。

2015-12-24的基础包：

> base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci usign kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-nf-conntrack kmod-nf-conntrack6 kmod-nf-ipt kmod-nf-ipt6 kmod-nf-nat kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libexpat libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script uboot-envtools

2014-09-01获取的基础包：

> base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci kmod-crypto-aes kmod-crypto-arc4 kmod-crypto-core kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-ipt-nathelper kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script uboot-envtools

默认包要包含在PACKAGES命令行参数中，并再加上必要的包：

```
luci-ssl ipset wget shadowsocks-libev iptables-mod-nat-extra bind-dig
```

如果你的openWrt版本是 ATTITUDE ADJUSTMENT，可能加上iptables-mod-nat-extra包，如果没安装的话iptables的端口转发会不支持。

注意，在编译前要把 shadowsocks-libev 及其他要用到的 .ipk 文件放到ImageBuilder的目录下packages/base/：

```
# 对于TLWR2543,DIR505A1:
~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64/packages/base/

# 对于WNDR4300:
~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/packages/base/
```

OpenWrt Image Builder的三个命令行参数

- PROFILE 指定设备类型，此处是 TLWR2543
- PACKAGES 指定要编译进固件的包
- FILES 指定要编译进固件的自定义文件，如网络有关配置文件, ~/Downloads/openwrt-tlwr2543

开始编译OpenWrt自动翻墙固件

```
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64
make image PROFILE=TLWR2543 PACKAGES="base-files busybox dnsmasq dropbear firewall fst
ools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci usign k
mod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-conntrack
kmod-ipt-core kmod-ipt-nat kmod-nf-conntrack kmod-nf-conntrack6 kmod-nf-ipt kmod-nf-ip
t6 kmod-nf-nat kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hot
plug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath kmod-ath9k kmod-ath9k-common kmod-
cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libexpat libiwinfo
 libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhc
p6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script uboot-envtools ipset
 wget shadowsocks-libev-polarssl luci-ssl" FILES=~/Downloads/openwrt-tlwr2543
```

编译好的的固件在ImageBuilder的bin/ar71xx/目录下。

```
# 对于TLWR2543,DIR505A1:
~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64/bin/ar71xx/

# 对于WNDR4300:
~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/bin/ar71xx/
```

升级固件要用到的是 ...sysupgrade.bin，比如 openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin

然后把这个固件刷进TP-LINK WR2543N，重启路由器后后就能免设置智能翻墙。

刷翻墙固件后管理员登录OpenWrt

刷好固件并重启路由器后，电脑连上无线网络 eastking-tlwr2543, 然后就可用密码 `fanqiang` 登录路由器。

- ssh登录openwrt管理路由器：

  ```
  ssh root@192.168.1.1
  ```

- 浏览器打开192.168.1.1登录

以后玩OpenWrt出问题，可以重新刷上这个翻墙固件就又可以在网上畅行无阻了。

参考：

- http://wiki.openwrt.org/doc/howto/obtain.firmware.generate
- https://wiki.openwrt.org/doc/howto/build

如何使用别人预编译的OpenWrt翻墙固件 for TP-LINK WR2543N (包含shadowsocks-libev)

如果你的无线路由器和我的一样，也是 TP-LINK wr2543N v1，你不想自己编译固件，那么可以下载我预先编译好的固件，刷好固件好，稍微设置下，就可以自动翻墙。

在下载和刷OpenWrt固件前，确保熟悉本教程的前面部分，已经配置好shadowsocks-libev服务端，并能自由进入路由器的安全模式。再次强调，刷机有风险，风险自承担。

该固件只是在OpenWrt trunk版加上：luci-ssl wget shadowsocks-libev的最新版，还有翻墙要用到的配置，没有添加其他任何内容。

翻墙默认配置

- 教程用到的OpenWrt翻墙配置文件](https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt
- 教程中用到的shadowsocks服务端配置文件

下载OpenWrt固件 for TP-LINK wr2543N

到下面的网址下载：

https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/

下载后保存在Ubuntu: `~/Downloads/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin`

复制OpenWrt固件到路由器

```
scp ~/Downloads/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

登录OpenWrt路由器,并查看文件大小是否正确

```
ssh root@192.168.1.1
root@OpenWrt: cd /tmp/
ls
```

升级OpenWrt固件(不保留原来配置)

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgra
de.bin
```

路由器重启后，电脑连接到无线网络 eastking-wr2543

ssh登录并修改设置：

```
ssh root@192.168.1.1
```

输入密码 `fanqiang` 登录

有时会提示错误：

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
cf:c5:12:34:56:0b:4d:1c:56:48:6a:87:04:cf:b8:83.
Please contact your system administrator.
Add correct host key in /home/openwrt-fanqiang/.ssh/known_hosts to get rid of this mes
sage.
Offending RSA key in /home/openwrt-fanqiang/.ssh/known_hosts:3
  remove with: ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
RSA host key for 192.168.1.1 has changed and you have requested strict checking.
Host key verification failed.
```

解决办法就是复制并运行提示中的清理命令：

```
ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
```

以下设置必须修改：

- /etc/shadowsocks.json
    - server必须改成你的服务器实际IP
- /etc/config/network
    - wan-username 和 wan-password必改
- /usr/bin/shadowsocks-firewall
    - 1.0.9.8必须改成你的服务器实际IP

如果你还改了其他默认值，请自行修改相应文件。不建议修改其他默认值，以提高一次成功率。

执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/network restart
```

测试一下是否可以在网上畅行无阻了。

本教程已经在github开源，欢迎提交改进，报告bug:

https://github.com/softwaredownload/openwrt-fanqiang

网件Netgear WNDR4300刷OpenWrt翻墙教程

网件Netgear WNDR4300是很多网友推荐的可刷OpenWRT的无线路由器。

WNDR4300有v1和v2的区别，目前国行都是v1版本。

eastking Status System Network Logout

Status

System

Hostname	eastking
Model	NETGEAR WNDR4300
Firmware Version	OpenWrt Designated Driver r47929 / LuCI (git-15.351.05963-967bb1f)
Kernel Version	4.1.13
Local Time	Tue Dec 22 10:39:19 2015
Uptime	13h 12m 4s
Load Average	0.08, 0.04, 0.05

Memory

Total Available	91060 kB / 125200 kB (72%)
Free	87472 kB / 125200 kB (69%)
Buffered	3588 kB / 125200 kB (2%)

网件Netgear WNDR4300无线路由器的优点

- 刷OpenWrt方便。购买后，登录管理界面可以直接刷OpenWrt
- WNDR4300自带不死uboot，刷机比较安全
- 硬件配置高。据网友测试，同时接入40台机器都没有问题
- 无线信号强。150平方的室内基本无信号死角
- 有一个 USB 接口，可以挂载设备

网件Netgear WNDR4300国行硬件信息

千兆双频，300+450Mbps的无线连接，2.4G和5G无线信号可以同时使用，1000Mbps有线端口，内置5天线（两根5G+三根2.4G），采用Atheros AR9344处理器，频率550MHz，128M DDR2内存，128M ROM，USB可接硬盘进行共享，带有wifi开关按钮可以单独关闭无线信号。

```
Version        v1
CPU            Atheros AR9344 rev2 560MHz MIPS 74Kc V4.12
Ram            128MiB
Flash        128MiB NAND
Network        1 WAN + 4x LAN (GBit)
Wireless    AR9580 [an 3x3:3] + AR9344 [bgn 2x2:2]
USB            Yes
```

如何购买网件Netgear WNDR4300无线路由器

目前自营电商的价格一般是299元，TB价大约280元。

参考信息

- Netgear WNDR4300 OpenWrt官网Wiki
- Windows下Netgear WNDR4300刷OpenWrt固件PDF教程 by 书浅
- gy408预编译集成固件for WNDR4300

最简单的路由器刷OpenWrt固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读OpenWrt翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

下载和设置OpenWrt Image Builder for 网件Netgear WNDR4300路由器

下载OpenWrt ImageBuilder时有二种选择：稳定发行版和最新trunk版。

OpenWrt ImageBuilder for 网件Netgear WNDR4300稳定发行版的下载

进入网址： http://downloads.openwrt.org/

现在是2015年12月22日，可以看到：

> Chaos Calmer 15.05
> Released: Fri, 11 Sep 2015

WNDR4300是NAND内存，进入下面的网址下载适合WNDR4300的ImageBuilder稳定发行版:

http://downloads.openwrt.org/chaos_calmer/15.05/ar71xx/nand/

我尝试用稳定发行版编译自动翻墙固件，出现错误，后来改用trunk版就顺利成功了。本教程用的是trunk版。

OpenWrt ImageBuilder for 网件Netgear WNDR4300最新trunk版的下载

Linux下，下载工具一般默认保存到 ~/Downloads，工作在Downloads目录，下载，解压和编译也比较方便。

```
cd ~/Downloads
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/nand/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64.tar.bz2
tar -xjf OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64.tar.bz2
```

网件Netgear WNDR4300路由器完全使用128M内存教程

将ubi和firmware增加96M，完全使用128M flash,以实现 WNDR4300路由器 overlay分区大于90MB的功能

在linux下用vi命令可以很方便地查找和修改特定字符。

- 查找23552k，替换成121856k
- 查找25600k，替换成123904k

下面就用vi来修改：

```
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/target/linux/ar71xx/image
cp Makefile Makefile.bak

vi Makefile

#change ubi size to 121856k
# search
/23552k
# delete word
dw
# insert
i
121856k

#change firmware size to 123904k
/25600k
dw
i
123904k

#Save and exit
ZZ
```

修改好后是这样的：

```
wndr4300_mtdlayout=mtdparts=ar934x-nfc:256k(u-boot)ro,256k(u-boot-env)ro,256k(caldata),512k(pot),
2048k(language),512k(config),3072k(traffic_meter),2048k(kernel),121856k(ubi),123904k@0x6c0000(fir
mware),256k(caldata_backup),-(reserved)
```

确定网件Netgear WNDR4300路由器的PROFILE值

```
cd OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64
make info
```

下图最上面一行显示，PROFILE值是WNDR4300：

```
WNDR4300:
        NETGEAR WNDR3700v4/WNDR4300
        Packages: kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ledtrig-usbdev
```

确定应该包含在自编译WNDR4300路由器翻墙固件里的包

1. 在Linux下运行命令自动获取基础包：

```
echo $(wget -qO - http://downloads.openwrt.org/snapshots/trunk/ar71xx/nand/config | se
d -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1/ip')
```

结果如下：

> base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci usign kmod-lib-crc-ccitt kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-nf-conntrack kmod-nf-conntrack6 kmod-nf-ipt kmod-nf-ipt6 kmod-nf-nat kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-spi-bitbang kmod-spi-gpio kmod-ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libexpat libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script procd-nand ubi-utils uboot-envtools

2. 获取网件Netgear WNDR4300路由器相关包：

```
make info
Current Target: "ar71xx (Generic devices with NAND flash)"
Default Packages: base-files libc libgcc busybox dropbear mtd uci opkg netifd fstools
kmod-gpio-button-hotplug swconfig kmod-ath9k wpad-mini uboot-envtools dnsmasq iptables
 ip6tables ppp ppp-mod-pppoe firewall odhcpd odhcp6c
Available Profiles:
...
WNDR4300:
    NETGEAR WNDR3700v4/WNDR4300
    Packages: kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ledtrig-usbdev
```

那就再增加上面的Default Packages和WNDR4300 Packages。去重排序后，再去掉dnsmasq，

3. 增加自定义包

```
ipset wget libopenssl shadowsocks-libev luci-ssl iptables-mod-nat-extra bind-dig dnsma
sq-full
```

Dnsmasq 提供 DNS 缓存和 DHCP 服务功能。作为域名解析服务器(DNS)，dnsmasq可以通过缓存 DNS 请求来提高对访问过的网址的连接速度。作为DHCP 服务器，dnsmasq 可以为局域网电脑提供内网ip地址和路由。

默认的dnsmasq为base版本，该版本不能对特定的域名地址进行标记操作（因为我们需要对一些特定域名如twitter等进行标记），改为更加强大的dnsmasq-full

luci-ssl是用来网页界面管理路由器，安装后就可以 https://192.168.1.1 登录WNDR4300路由器

bind-dig可以调试域名解析

shadowsocks-libev 翻墙主角

编译shadowsocks-libev ipk for网件Netgear WNDR4300路由器

不同OpenWrt版本下编译的shadowsocks-libev ipk一般是不能通用的。比如现在用的是trunk版的OpenWrt，如果使用OpenWrt Chaos Calmer 15.05 下编译的shadowsocks-libev，可能安装后根本不能启动。

前面我曾编译出翻墙固件，其中shadowsocks-libev是别人编译，从sourceforge上下载的，刷上固件后，shadowsocks总是没有自动启动，运行/usr/bin/ss-redir，报告没有找到这个文件，其实文件是在的，只是不兼容。所以，最好还是自行编译shadowsocks-libev。

按官网的说法，以下 不要使用**root**用户来操作

编译shadowsocks-libev ipk安装包

请同时参考前面的教程：编译shadowsocks-libev for OpenWrt ipk安装包

下面都是在Linux下操作。

```
cd ~/Downloads
git clone git://git.openwrt.org/openwrt.git

pushd package
git clone https://github.com/shadowsocks/shadowsocks-libev.git
popd

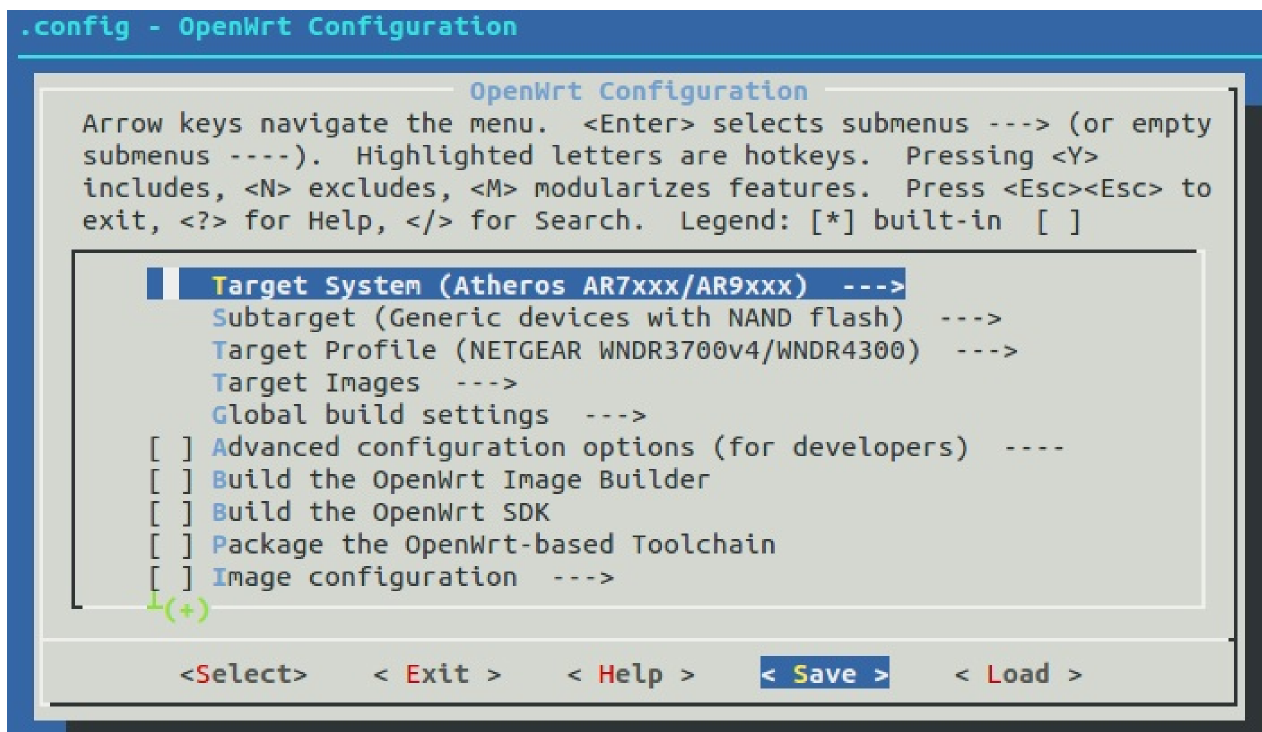
cd ~/Downloads/openwrt
./scripts/feeds update -a
./scripts/feeds install -a

make defconfig
make prereq
make menuconfig

# Target System: Atheros AR7xxx/AR9XXX
# Subtarget: Generic device with NAND flash
# Target Profile: (因我们只是编译包，这步可以不选)
# Network, 选择shadowsocks-libev-openssl 和 shadowsocks-libev-polarssl,  按m设置为编译独立ipk安装包
# Save && Exit

#  这一步花了几个小时
make tools/install && make toolchain/install

# 开始编译
make V=99 package/shadowsocks-libev/openwrt/compile
```



输出文件在 openwrt/bin/ar71xx/packages/base/目录下，主要有：

```
shadowsocks-libev_2.4.3_ar71xx.ipk
shadowsocks-libev-polarssl_2.4.3_ar71xx.ipk
libopenssl_1.0.2e-1_ar71xx.ipk
libpolarssl_1.3.15-1_ar71xx.ipk
```

把所有ipk都复制到ImageBuilder的packages/base目录下

```
cd ~/Downloads/openwrt/bin/ar71xx/packages/base/
cp * ~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/packages/base
```

设置网件Netgear WNDR4300翻墙配置文件

要翻墙成功，这一步是最重要的。

分三步，下载本项目openwrt-fanqiang；复制配置文件；修改配置文件。

下面以linux系统 ~/Downloads 下操作为例。

下载包含默认翻墙配置文件的openwrt-fanqiang项目

- git下载openwrt-fanqiang项目

  cd ~/Downloads git clone https://github.com/softwaredownload/openwrt-fanqiang

- 或者下载zip文件

  https://github.com/softwaredownload/openwrt-fanqiang/archive/master.zip

本地项目文件夹是： ~/Downloads/openwrt-fanqiang

复制openwrt-fanqiang里面的翻墙配置文件到openwrt-wndr4300目录下

建立一个配置文件夹，以路由器型号结束，如 ~/Downloads/openwrt-wndr4300。

```
cd ~/Downloads
mkdir openwrt-wndr4300

cd openwrt-fanqiang
cp -R openwrt/default/* ~/Downloads/openwrt-wndr4300/
cp -R openwrt/wndr4300/* ~/Downloads/openwrt-wndr4300/
```

上面的操作，先复制共用的配置文件 openwrt/default/ 到 *openwrt-wndr4300*目录下
然后复制*wndr4300*专用的配置文件到 *openwrt/wndr4300/* 到 openwrt-wndr4300目录下，如果有同名文件就覆盖。

如果你要贡献本项目，也是先在openwrt-fanqiang/openwrt目录下先建立路由器型号为名称的文件夹，再把专用的配置文件放到此文夹下。注意文件夹和文件名都是小写的。

修改Netgear WNDR4300翻墙配置文件

主要修改以下文件：

```
openwrt-wndr4300/etc/shadowsocks.json
openwrt-wndr4300/usr/bin/shadowsocks-firewall
openwrt-wndr4300/etc/uci-defaults/defaults
```

为了方便以后升级，可以写个bash文件自动修改配置文件。

一切操作尽量自动化，你甚至可以自动化一切操作：下载ImageBuilder，下载OpenWrt源码，下载shadowsocks-libev源码，同步openwrt-fanqiang源码，编译ipk，修改翻墙设置，编译翻墙固件，早上一觉醒来，新鲜出炉、美味可口的翻墙固件就已经摆放在桌上了。

下面是一个自动修改配置文件的例子，从中可以知道需要修改哪些地方。从2015年12月起，可能用于自动化修改的默认值都应该标准化，方便自动化操作。

config-wndr4300.sh:

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date:   2015-12-20

REPOSITORY=~/Downloads/openwrt-fanqiang
CONFIG=~/Downloads/openwrt-wndr4300

createdir() {
    rm -rf $CONFIG
    mkdir $CONFIG
}

copy() {
    cp -R $REPOSITORY/openwrt/default/* $CONFIG/
    cp -R $REPOSITORY/openwrt/wndr4300/* $CONFIG/
}

setmod() {
    chmod +x $CONFIG/usr/bin/shadowsocks-firewall
    chmod +x $CONFIG/etc/uci-defaults
    chmod +x $CONFIG/etc/uci-defaults/*
}

modify() {
    # server ip address
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/etc/shadowsocks.json
```

```
    # server_port
    sed -i 's/1098/server_port/' $CONFIG/etc/shadowsocks.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/etc/shadowsocks.json

    # password
    sed -i 's/killgfw/killgfw/' $CONFIG/etc/shadowsocks.json

    # method
    sed -i 's/aes-256-cfb/aes-256-cfb/' $CONFIG/etc/shadowsocks.json


    # server ip addresss
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/usr/bin/shadowsocks-firewall

    # local_port
    sed -i 's/7654/7654/' $CONFIG/usr/bin/shadowsocks-firewall


    # ppoe username
    sed -i 's/wan-username/wan-username/' $CONFIG/etc/uci-defaults/defaults

    # ppoe password
    sed -i 's/wan-password/wan-password/' $CONFIG/etc/uci-defaults/defaults

    # wifi password
    sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/etc/uci-defaults/defaults

    # router login password for root
    sed -i 's/\\nfanqiang/\\nfanqiang/' $CONFIG/etc/uci-defaults/defaults
}



if [ "$1" = "createdir" ]; then
    createdir
elif [ "$1" = "copy" ]; then
    copy
elif [ "$1" = "setmod" ]; then
    setmod
elif [ "$1" = "modify" ]; then
    modify
else
    echo "usage: createdir copy setmod modify"
fi
```

config-wndr4300.sh使用方法：

必改值是：

```
server_ip
wan-username
wan-password
```

如果你比较懒，就改这三项就行了，可以说本教程是最简单的翻墙方案了。

选改值：

```
router login password for root
wifi password
```

其他值一般保持默认值就可以了。

假设config-wndr4300.sh在~/Downloads目录下，运行命令自动修改翻墙配置：

```
cd ~/Downloads
sudo chmod +x config-wndr4300.sh
./config-wndr4300.sh createdir
./config-wndr4300.sh copy
./config-wndr4300.sh setmod
./config-wndr4300.sh modify
```

编译OpenWrt自动翻墙固件 for 网件Netgear WNDR4300路由器

经过前面几个步骤，一切准备就绪，下面就正确开始编译Netgear WNDR4300专用全自动翻墙固件了。

编译OpenWrt自动翻墙固件前的系统准备

```
sudo apt-get update
sudo apt-get install git-core build-essential libssl-dev libncurses5-dev unzip
```

OpenWrt Image Builder的三个命令行参数

- PROFILE 指定设备类型，此处是 WNDR4300
- PACKAGES 指定要编译进固件的包
- FILES 指定要编译进固件的自定义文件，如网络有关配置文件, 默认目录：~/Downloads/openwrt-wndr4300

开始编译OpenWrt自动翻墙固件 for 网件Netgear WNDR4300路由器

命令：

```
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/

make image PROFILE=WNDR4300 PACKAGES="base-files busybox dropbear firewall fstools jso
nfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci usign kmod-lib-
crc-ccitt kmod-ipv6 kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-
nf-conntrack kmod-nf-conntrack6 kmod-nf-ipt kmod-nf-ipt6 kmod-nf-nat kmod-ppp kmod-ppp
oe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-spi-bitbang kmod-spi-gpio kmod-a
th kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtabl
es libblobmsg-json libexpat libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6t
ables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jsh
n libjson-script procd-nand ubi-utils uboot-envtools base-files libc libgcc busybox dr
opbear mtd uci opkg netifd fstools kmod-gpio-button-hotplug swconfig kmod-ath9k wpad-m
ini uboot-envtools kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ledtrig-usbdev ipset wge
t libopenssl shadowsocks-libev luci-ssl iptables-mod-nat-extra bind-dig dnsmasq-full"
FILES=~/Downloads/openwrt-wndr4300
```

编译时报错：

> opkg_install_cmd: Cannot install package kmod-ipv6

移除 kmod-ipv6后编译成功。

编译好的的固件在：

```
~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/bin/ar71xx/
```

其中包含：

```
openwrt-ar71xx-nand-wndr4300-ubi-factory.img
openwrt-ar71xx-nand-wndr4300-squashfs-sysupgrade.tar
```

可见生成了二种格式的固件，img 格式和 tar 格式。 其中 img 格式只能用 tftp 的方法进行刷入。而 tar 也只能通过 已刷了Openwrt的WEB端进行刷入。

部分编译错误处理

1. Build dependency: Please install the openssl library (with development headers)

   For Centos：

   ```
   yum install openssl-devel
   ```

   For Ubuntu：

```
sudo apt-get install libssl-dev
```

2. Unable to open feeds configuration in line 42

   使用 `svn co svn://svn.openwrt.org/openwrt/trunk/` 下载后再编译的方法没有遇到这个问题。

3. configure: error: you should not run configure as root (set FORCE_UNSAFE_CONFIGURE=1 in environment to bypass this check)

   See config.log' for more details

   将下载的文件的所有者改为自己,假设用户名是ubuntu

   `sudo chown -Rv ubuntu /home/ubuntu/openwrt`

   再重新运行`make`

网件Netgear WNDR4300路由器怎样刷OpenWrt自动翻墙固件

两种翻墙固件格式 img tar的区别

```
openwrt-ar71xx-nand-wndr4300-ubi-factory.img
openwrt-ar71xx-nand-wndr4300-squashfs-sysupgrade.tar
```

我们编译出了两种固件，一种为 ...ubi-factory.img 格式，一种为 ...squashfs-sysupgrade.tar 格式。 其中 img 格式只能用 tftp 的方法刷入。而 tar 只能通过已刷了Openwrt的WEB端进行刷入。下面分别说明 两种不同的刷入方法：

tftp刷固件的方式，不管原来的固件是什么格式，都可以刷factory.img

网件Netgear WNDR4300路由器进入恢复模式的方法

- 关闭路由器电源
- 用 牙签，或其他尖物 按住设备背面的机身背面的红色小圆孔(Restore Factory Settings button)
- 开启电源开关
- 观察电源灯（此时保持按住Restore Factory Settings按钮不要松手），直到电源灯由 橙色闪烁 状态变到 绿色闪烁 状态（说明设备已经进入到了 TFTP修复模式 ）

Linux下Netgear WNDR4300路由器用tftp刷翻墙固件

- 将电脑用网线连接到设备的 LAN口，而不是wan口。国行Netgear WNDR4300的wan口是黄色的
- 将电脑的本地连接IP设置为 192.168.1.X （此例中IP地址设置为 192.168.1.2 ），子网掩码为 255.255.255.0，网关为192.168.1.1
- 路由器进入恢复模式
- 测试能否连接到路由器：

```
ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
Warning: time of day goes back (-3646479862160196504us), taking countermeasures.
Warning: time of day goes back (-3646479862160196420us), taking countermeasures.
```

- 网件Netgear WNDR4300路由器刷翻墙固件

```
sudo apt-get install tftp
# 进入固件所在目录
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/bin/ar71xx
echo -e "binary\nrexmt 1\ntimeout 60\ntrace\nput openwrt-ar71xx-nand-wndr4300-ubi-factory.img\n" | tftp 192.168.1.1
```

- 观察指示灯，文件传送完毕后，等待80秒左右，设备会自动重启（请耐心等待，切勿将路由器手动断电）。设备重启后，看到亮绿灯，一定要按机身后面的电源开关手动断电、开机，否则可能没有无线5G 这不是BUG，其他openwrt也是一样的。每次刷factory.img都要这样

Windows下Netgear WNDR4300路由器用tftp刷翻墙固件

- 启用tftp。Windows 10下：控制面板，所有控制面板项，程序和功能，启用或关闭Windows功能，启用“TFTP”客户端
- 将电脑用网线连接到设备的 LAN口
- 将电脑的本地连接IP设置为 192.168.1.X （此例中IP地址设置为 192.168.1.2 ），子网掩码为 255.255.255.0，网关192.168.1.1
- 路由器进入恢复模式
- 测试能否连接到路由器： ping 192.168.1.1
- 网件Netgear WNDR4300路由器刷翻墙固件
    - 按Windows+R,输入cmd并回车调出命令行程序
    - 假设openwrt-ar71xx-nand-wndr3700v4-ubi-factory.img在C:\盘
    - 运行命令：

    ```
    cd C:\
    tftp -i 192.168.1.1 put openwrt-ar71xx-nand-wndr3700v4-ubi-factory.img
    ```

    ```
    C:\>tftp -i 192.168.1.1 put openwrt-ar71xx-nand-wndr4300-ubi-factory.img
    Transfer successful: 6815873 bytes in 3 second(s), 2271957 bytes/s
    ```

- 观察指示灯，设备重启后，看到亮绿灯，再手动断电、开机，否则可能没有无线5G

更详细的WNDR4300刷openwrt PDF图文教程：

Windows下Netgear WNDR4300刷OpenWrt固件PDF教程 by 书浅

登录并设置已经刷了OpenWrt 翻墙固件的网件Netgear WNDR4300路由器

Netgear WNDR4300 预编译翻墙固件下载(2015-12-22)

https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/

你按照本教程编译了WNDR4300路由器 OpenWrt 全自动翻墙固件，并且刷进了路由器，如果一切正常，就可以零设置自动翻墙了。运气不够好，就要登录路由器修改一下设置。

你懒得自己编译翻墙固件，下载了本教程提供的Netgear WNDR4300路由器翻墙固件并刷进了路由器，就必须手动修改一些值才能自动翻墙。

本教程就针对上面这两种情况。

怎样登录已经刷了OpenWrt 翻墙固件的网件Netgear WNDR4300路由器

用网线连接电脑和路由器，将电脑的本地连接IP设置为 192.168.1.2，子网掩码为 255.255.255.0，网关为：192.168.1.1

- 网页登录地址： https://192.168.1.1
- ssh登录： root @192.168.1.1
- 默认登录密码： fanqiang

Linux下ssh登录WNDR4300路由器并修改设置

```
eastking@ubuntu:~$ ssh root@192.168.1.1
root@192.168.1.1's password:
BusyBox v1.24.1 (2015-12-18 16:02:57 CET) built-in shell (ash)

Author:
https://github.com/softwaredownload/openwrt-fanqiang

# server_ip
root@OpenWrt:~# vi /etc/shadowsocks.json

# server_ip
root@OpenWrt:~# vi /usr/bin/shadowsocks-firewall

# wan-username, wan-password
root@OpenWrt:~# vi /etc/config/network

# wifi password, optional
root@OpenWrt:~# vi /etc/config/wireless
```

如果你修改了本教程默认的shadowsocks local_port和tunnel_port，还得修改/etc/dnsmasq.d/下相关文件中的端口号。

执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/network restart
```

D-Link DIR-505路由器刷OpenWrt固件翻墙教程

前面的教程用结合 TP-LINK TL-WR2543N 来讲解翻墙原理与方法，并不是我特别推荐TP-LINK TL-WR2543N，而是因为手头正好有这个路由器。毫无疑问，初学者使用教程同款路由器比较容易上手。但此型号趋向退市，价格也不便宜，网上有二手货，如果功能正常倒也可以考虑。

另外的选择，是使用 D-Link DIR-505 便携式路由器。配置高，价格便宜。

D-Link DIR 505 硬件信息

```
Architecture:        MIPS 24Kc
Vendor:                Atheros
Bootloader:            UBoot 1.1.4
System-On-Chip:      SoC: Atheros AR9330 rev 1
CPU/Speed:             Atheros AR9330 400.000MHz
Flash-Chip:            NANYA NT5TU32M16DG-AC
Flash size:            8192 KiB
RAM:                 64 MiB
Wireless:              802.11b/g/n
Ethernet:              10/100 full duplex
USB:                 Yes 1 x 2.0 ar7240-ehci
Serial:                Yes - tested working over TTL converter (3.3V!)
JTAG:     Nope
```

与之同价格档次的TP-LINK TL-WR706N 150M迷你型无线路由器， AR9331 SOC 2MB Flash/16MB RAM, 相比之下简直是垃圾。我花数百元购买的TP-LINK TL-WR2543N，也不过是8MB Flash, 64MB RAM内存。

还有，D-Link DIR-505 自带不死恢复模式，调试OpenWrt系统出现问题时我们既可以进D-Link的恢复模式刷新固件，也可以进入OpenWrt的恢复模式刷新固件，可谓是最安全的路由器。

如何购买 D-Link DIR 505 A1

我不是D-Link的员工，也无意为其做广告。DIR-505是我购买的第一款D-Link路由器。

我是2014年8月从淘宝 D-Link官方旗舰店买的 D-Link DIR 505 A1，69元, 固件版本号：1.03CN。买了后，看了下手机淘宝，只要59元。准备再入一个，都刷上OpenWrt，方便随时随地无障碍上网。

最简单的路由器刷OpenWrt固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读OpenWrt翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

如何进入 DIR-505 恢复模式

在学习OpenWrt可能要测试很多配置，有时会出现错误，需恢复或补救，这时就需要进入路由器的恢复模式。

有两种方法进入 DIR-505 的恢复模式。

进入D-Link 恢复模式

把 DIR-505和 计算机用网线连接起来，设置计算机网卡的IPv4地址为192.168.0.12， 子网掩码255.255.255.0， 在路由器启动时顶住reset孔，当红色指示灯开始缓慢闪烁时，松开reset孔。然后浏览器打开 192.168.0.1，这里你可以上传原厂固件或刷 OpenWrt 固件。

Plug in your computer to the unit, assign it an ip address of 192.168.0.12, and boot the unit up while holding down the reset. Once the red light starts to blink slowly, release the reset, and go to 192.168.0.1 on your web browser. From there you can upload a new image. After successful flashing, you'll see a "Success" page in your browser.

刷新固件完成后，重新改回自动获取IP地址。

进入 OpenWrt 恢复模式

用网线将路由器和电脑连接起来，将电脑网卡的IPv4地址设置成 192.168.1.23

路由器插上电源重新开机, 在启动时多次按压路由器侧面的圆形 WPS 按钮直到 LED 指示灯开始快速闪烁。

For the generic failsafe mode you can follow http://wiki.openwrt.org/doc/howto/generic.failsafe You can use the WPS button for that. While booting up, just press it several times until the LED flashes very quick. If you're still not able to telnet it on 192.168.1.1 maybe there's something wrong on the client-side.

接下来就是ubuntu 里 telnet 进入 OpenWrt 并设置 root 密码。

```
telnet 192.168.1.1
```

telnet连上后就设置root密码，自动启用 ssh:

```
root@OpenWrt:/# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@OpenWrt:/#
```

这里，可以在 Ubuntu 里 Ctrl + Shift + t新开一个命令行窗口，尝试 ssh 连接OpenWrt:

```
ssh root@192.168.1.1
```

如果 ssh 连上了，则后面设置的内容和 前面 TLWR-2543N 翻墙教程一样了。

要注意的是，D-Link DIR-505 使用接口名称 eth1 而不是通常的 eth0.

Other than similar routers (e.g., the TP-Link TL-WR703N), the D-Link DIR-505 uses the interface eth1 rather than eth0. This means that if you build your own firmware, you must configure /etc/config/network accordingly (option ifname 'eth1'), or you will not be able to connect later on via Ethernet.

如果 telnet 连不上，尝试一下直接ssh登录。

设置D-Link DIR-505k路由器无线连接

在没有设置无线连接前，要登录OpenWrt，必须用网线把电脑和路由器连接起来，不太方便。设置无线连接后，电脑就可以通过无线方式连上路由器，再登录 DIR-505 OpenWrt进行设置。

```
uci set wireless.@wifi-device[0].disabled=0;
uci set wireless.@wifi-iface[0].ssid='eastking-dir505';
uci set wireless.@wifi-iface[0].encryption='psk2+ccmp';
uci set wireless.@wifi-iface[0].key='icanfly9876';
uci commit wireless;
wifi
```

设置好无线连接后，就可以拔掉电脑的有线连接，连接无线，再ssh登录路由器。

参考：

- https://forum.openwrt.org/viewtopic.php?id=38742&p=8
- http://wiki.openwrt.org/toh/d-link/dir-505#debricking.
- http://my.oschina.net/umu618/blog/271630

D-Link DIR-505 A1 刷 OpenWrt固件过程

D-Link 路由器是锁区的, 不能直接刷OpenWrt 固件。要先到D-Link 官方国际站下载原厂固件，用16进制编辑器把DEF改成CN，升级固件，再刷OpenWrt固件。

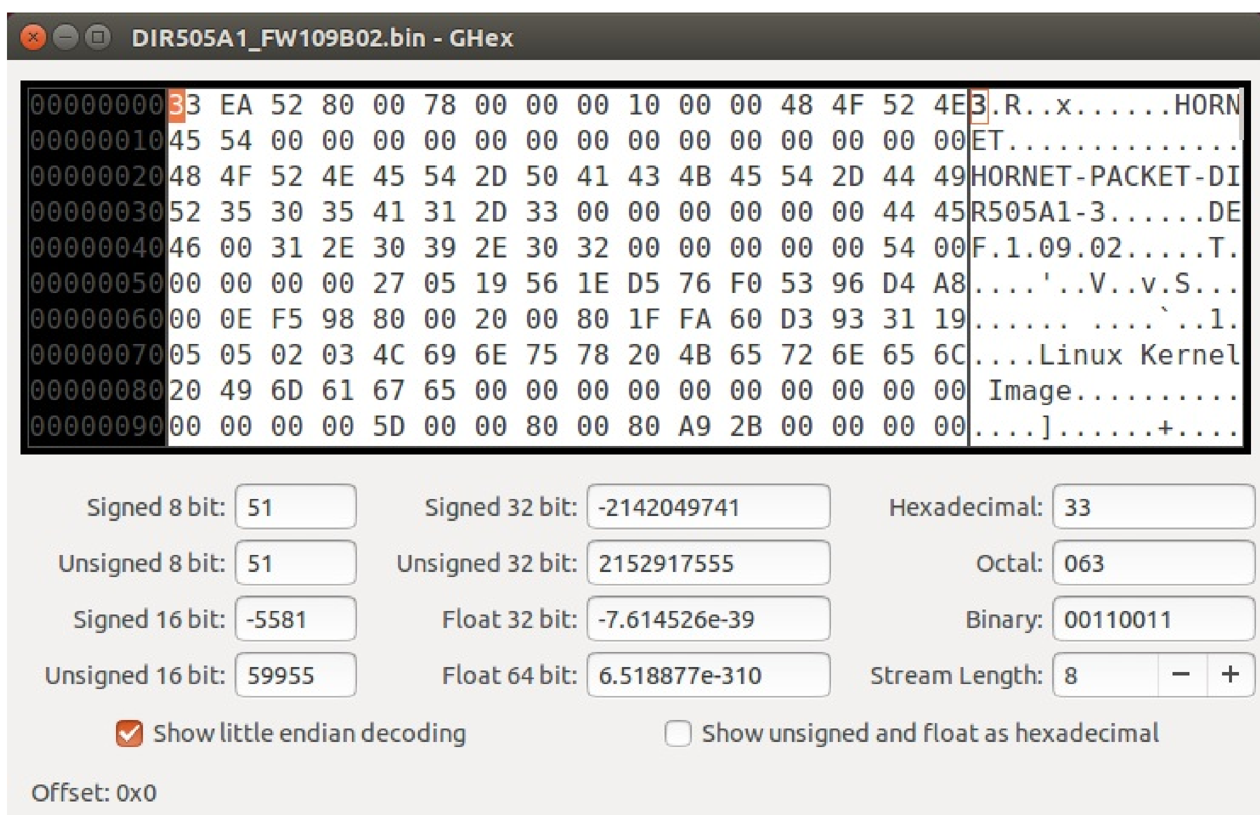
下载D-Link DIR-505 A1 国际版官方固件

下载地址：

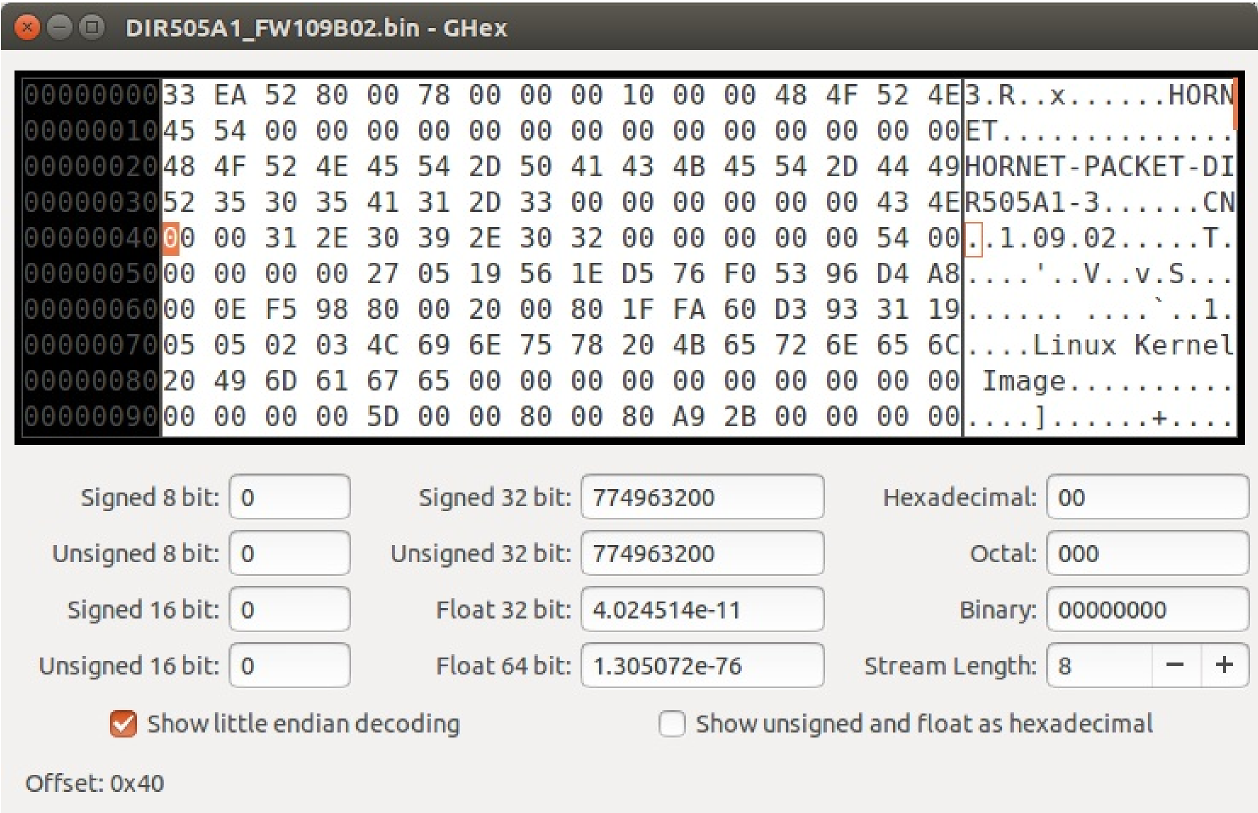
- http://support.dlink.com.au/download/download.aspx?product=DIR-505
- ftp://files.dlink.com.au/products/DIR-505/REV_A/Firmware/

用16进制编辑器修改固件的国家代码，DEF改成CN

准备一个16进制编辑器，在本文中，我用的是Ubuntu下的轻量级16进制编辑器GHex,把固件拖到GHex打开固件。



修改后变成如下：



Alt+S保存对固件的修改。

你也可以到下面网址直接下载修改好16进制值的固件：

https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/

刷修改国家后的官方固件

按照路由器官方手册，电脑连上路由器。

在Ubuntu下电脑连接DIR-505路由器的方法：　DIR-505路由器出厂默认设置没有开启DHCP，所以我们要给电脑手动设置和路由器同网段的IPv4地址才能连上路由器。 路由器插上电源。右上角无线信号处，选择 Edit Connections, 选择dlink-xxxx,xxxx为路由器MAC ID的后4位，Edit..., IPv4 Setings, Method选择 Manula手动，Address选择Add, 设置Address为192.168.0.2，Netmask 255.255.255.0, Gateway 192.168.0.1。如此设置好后电脑就能连上无线网络dlink-xxxx了。

浏览器首次进入 http://192.168.0.1 会出现设置向导，点取消，然后会出现密码登录页面：



直接点击 登入 按钮，再点击界面上部的 维护，然后点击左侧栏的 固件 进入升级固件页面，点击 **Browse...** 上传我们修改好的固件：



然后点击 上传 按钮完成刷新固件，接下来就可以刷 OpenWrt固件了。

DIR-505A1 刷 OpenWrt 固件

下载 OpenWrt 固件 for DIR-505A1

- http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/
- http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

DIR-505刷OpenWrt固件

我们是在原厂固件上刷 OpenWrt,一定要下载 factory.bin.上传后，等待150秒，DIR-505A1成功刷上了OpenWrt开源固件。

固件和语言包信息

当前固件版本：1.09　　日期：2014/June/10
当前语言包版本：1.00 CN　　日期：2012/5/8

网上查询产品固件和语言包最新版本：立刻检查

韧体升级

注意：某些固件升级会将设置复位至出厂默认设置。在进行升级前，请确认从维护→系统 界面保存当前配置

如要升级固件，您的计算机必须以有线方式接入AP，输入升级固件的文件名，然后点击上传按钮。

上传：

Browse... openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

上传

参考：

- http://my.oschina.net/umu618/blog/268466

D-Link DIR-505启用工作模式开关

DIR-505 硬件开启四种应用模式

D-Link DIR-505 在全球销售多款型号，不同型号外观不一样，但内部硬件是一样的。在中国销售的 DIR-505 A1, 也就是本教程所用的型号， 模式开关共有三档，在开关处动动手，就可以启用四种模式。

撕掉标贴，去掉螺丝，就可以打开DIR-505,把开关剪短，剪掉挡住开关上推的底面，完工后如下图：



Router模式和AP模式

便携式式无线路由器常有Router模式和AP模式，有的路由器用两个档位对应这两种模式，拨到Router档就用Router模式，拨到AP档就用AP模式。DIR-505 原厂固件，Router和AP共用一个档位，需要用哪种，需要登录路由器进行选择和设置。现在我们已经刷了OpenWrt，档位对应的模式需要自己定义设置。

在本教程中，把新开的第四档作为AP档，原来的Router/AP档作为Router档。

在Router模式时，DIR-505作为无线路由器使用，有线接口作为WAN口，连接到ADSL Modem。计算机通过无线的方式连接到路由器。在这种模式下一般需要设置拨号上网帐号。

在AP模式时，通常在DIR-505前端还有路由器，DIR-505的有线接口作为LAN口使用，前端路由器的LAN口引出网线连接到DIR-505. 在宾馆上网，把有线扩展为无线常应用此种模式。

/etc/rc.local 利用 GPIO 读取开关位置

rc.local内容如下：

```
# Put your custom commands here that should be executed once
# the system init finished. By default this file does nothing.

if [ ! -f /etc/config/backup/network ]; then
        cp /etc/config/network /etc/config/backup/
        cp /etc/config/wireless /etc/config/backup/
        cp /etc/config/firewall /etc/config/backup/
        cp /etc/config/dhcp /etc/config/backup/
fi


read_gpio() {
 (echo $1 > /sys/class/gpio/export) >& /dev/null
 (echo "in" > /sys/class/gpio/gpio$1/direction) >& /dev/null
 return `cat /sys/class/gpio/gpio$1/value`;
}
read_gpio 19;
v=$?;
read_gpio 20;
v=$v$?;
read_gpio 21;
v=$v$?;
read_gpio 22;
v=$v$?;
read_gpio 23;
v=$v$?;
case "$v" in
 10001) v="router";;
 11001) v="repeater";;
 01001) v="hotspot";;
 11000) v="ap";;
 *) v="error";;
esac

/usr/bin/$v

logger working mode: $v


exit 0
```

原理：先备份原始的配置文件，不同模式的设置都是基于原始配置文件，以免出现混乱。

在/usr/bin目录下创建相应模式的bash文件，根据不同的GPIO值调用的不同的文件。 在本教程中中主要应用 /usr/bin/router和 /usr/bin/ap这两个文件。

代码的最新版本，请查看:

- https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/dir505

你使用时，可以直接下载整个项目到本地，所有配置文件自然在其中:

```
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

参考：

- http://my.oschina.net/umu618/blog/273945

DIR-505 Router 模式

/usr/bin/router 代码：

```
#!/bin/sh

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2014-08-22

cp /etc/config/backup/* /etc/config/

uci delete network.lan.ifname
uci delete network.lan.type

uci add network interface
uci rename network.@interface[-1]='wan'
uci commit network

uci set network.wan.ifname='eth1'
uci set network.wan.peerdns=0
uci set network.wan.proto='pppoe'
uci set network.wan.username='wan-username'
uci set network.wan.password='wan-password'
uci set network.wan.peerdns=0

uci commit network

# default is no this option
#uci set dhcp.lan.ignore=0
#uci commit dhcp

uci set wireless.@wifi-device[0].channel=11
uci set wireless.@wifi-device[0].txpower=15
uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-device[0].country='CN'
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking-dir505'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='icanfly9876'

uci commit wireless
wifi

/etc/init.d/network restart
```

代码说明：

先把备份的原始配置文件覆盖到配置文件目录，所有设置都基于原始配置文件。 在使用 Router 模式时，有线接口为WAN口，这时wan的 interface name 是 eth1, 默认lan的 interface name 使用了 eth1, 要删除。

代码的最新版本，请查看:

- https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/dir505

DIR-505 AP 模式翻墙教程

/usr/bin/ap 代码：

```
#!/bin/sh

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2014-08-22

cp /etc/config/backup/* /etc/config/

uci set network.lan.gateway=192.168.1.1
uci set network.lan.dns=192.168.1.1
uci set network.lan.ipaddr=192.168.1.234

uci commit network

uci set dhcp.lan.ignore=1
uci commit dhcp

uci set wireless.@wifi-device[0].channel=11
uci set wireless.@wifi-device[0].txpower=15
uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-device[0].country='CN'
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking-dir505'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='icanfly9876'

uci commit wireless
wifi

/etc/init.d/network restart
```

代码说明：

在AP模式下，DIR-505的有线接口作为LAN口使用，连接到前端路由器的LAN口。 假设DIR-505前端路由器的IP地址是192.168.1.1，设置DIR-505的lan 网关和dns都是192.168.1.1，再设置 DIR-505的 lan IP地址为192.168.1.234。

DIR-505穿越功夫网翻墙方法

假设上级路由器没有设置翻墙

电脑设置无线连接 eastking-dir505 的IPv4地址是 192.168.1.235（不同于路由器的地址），设置子网掩码为255.255.255.0，网关和DNS为路由器的地址即192.168.1.234，重启路由器后，电脑连上 eastking-dir505　即可自动翻墙

原理：以DIR-505作为DNS服务器，我们已经把DIR-505设置成翻墙路由器，自然可以打败功夫网了。

假设上级路由器已经翻墙

电脑设置无线连接 eastking-dir505为DHCP即可。 原理：以上级路由器作为DNS服务器，上级路由器已经翻墙，二级路由器就可以免设置自动翻墙了。

如果你想节省路由器资源，这时就可以禁用 dir-505 dns及翻墙相关服务：

```
/etc/init.d/dnsmasq stop
/etc/init.d/dnsmasq disable
/etc/init.d/shadowsocks stop
/etc/init.d/shadowsocks disable
```

代码的最新版本，请查看:

- https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/dir505

参考：

- http://wiki.openwrt.org/doc/recipes/bridgedap

编译OpenWrt全自动翻墙固件 for D-Link DIR-505 A1

除了增加模式转换开头，其他和编译　TP-LINK WR2543N翻墙固件一样。

下载适合D-Link DIR505无线路由器的Image Builder

Image Builder又叫Image Generator，利用它我们可以方便地定制适合自己无线路由器的固件。

选择 OpenWrt版本：

```
* 进入  http://downloads.openwrt.org/
* Development Snapshots 最新开发版，我们的选择
* Binary Releases, 最后发行的稳定版本
* 进入 http://downloads.openwrt.org/snapshots/trunk/
```

选择 CPU 类型：

```
* 选择 ar71xx: http://downloads.openwrt.org/snapshots/trunk/ar71xx/
```

选择 Flash 类型：

```
* 选择 generic: http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/
```

下载 Image Builder for DID-505 (Version: 23-Dec-2015 16:38)

```
* 页面搜索 dir-505 如果找到，说明我们找对了目录
* 下载 Image Builder:

cd ~/Downloads
wget http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64.tar.bz2
tar -xjf  OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64.tar.bz2
```

确定OpenWrt无线路由器的PROFILE值

```
cd OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64
make info
```

找到自己固件的型号，D-Link DIR 505 A1的PROFILE值是DIR505A1。如下图：

```
DIR505A1:
        D-Link DIR-505 rev. A1
        Packages: kmod-usb-core kmod-usb2 kmod-ledtrig-usbdev
DIR600A1:
        D-Link DIR-600 rev. A1
```

找出默认应该包含进OpenWrt固件的包

对于D-Link DIR-505 A1 无线路由器来说，可以这样获取：

```
echo $(wget -qO - http://downloads.openwrt.org/snapshots/trunk/ar71xx/generic/config |
 sed -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1/ip')
```

2015-12-24的基础包：

> base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci usign kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-nf-conntrack kmod-nf-conntrack6 kmod-nf-ipt kmod-nf-ipt6 kmod-nf-nat kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libexpat libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script uboot-envtools

2014-09-01查询得到的基础包是：

> base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci kmod-crypto-aes kmod-crypto-arc4 kmod-crypto-core kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-ipt-nathelper kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script uboot-envtools

默认包要包含在PACKAGES命令行参数中，并再加上必要的包：

```
luci-ssl ipset wget shadowsocks-libev iptables-mod-nat-extra bind-dig
```

注意，在编译前要把 shadowsocks-libev 及其他要用到的 .ipk 文件放到目录下：

```
~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64/packages/base/
```

如果你的openWrt版本是 ATTITUDE ADJUSTMENT，可能加上iptables-mod-nat-extra包，如果没安装的话iptables的端口转发会不支持。

按照教程 编译shadowsocks-libev for OpenWrt ipk安装包

下载和设定自定义翻墙配置文件

下面以linux系统 ~/Downloads 下操作为例。

```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

本地项目文件夹是： ~/Downloads/openwrt-fanqiang

建立一个配置文件夹，以路由器型号结束，如 ~/Downloads/openwrt-dir505。

```
cd ~/Downloads
mkdir openwrt-dir505

cd openwrt-fanqiang
cp -R openwrt/default/* ~/Downloads/openwrt-dir505/
cp -R openwrt/dir505/* ~/Downloads/openwrt-dir505/
```

上面的操作，先复制共用的配置文件 openwrt/default/ 到 *openwrt-dir505*目录下
然后复制*dir505*专用的配置文件到 *openwrt/dir505/* 到 openwrt-dir505目录下，如果有同名文件就覆盖。

设置可执行权限

```
cd ~/Downloads/openwrt-dir505
chmod +x usr/bin
chmod +x usr/bin/*
chmod +x etc/uci-defaults
chmod +x etc/uci-defaults/defaults
```

说明：etc/uci-defaults目录下的文件会在路由器第一次启动时执行一次。在这里我们设置一些常用值。

必须修改的DIR505翻墙配置文件：

- ~/Downloads/openwrt-dir505/etc/shadowsocks.json
  - server改成你的服务器实际IP
- ~/Downloads/openwrt-dir505/usr/bin/router
  - wan-username 和 wan-password改成实际值
- ~/Downloads/openwrt-dir505/usr/bin/shadowsocks-firewall
  - 1.0.9.8必须改成你的服务器实际IP

自动复制和修改DIR-505翻墙设置文件

config-dir505.sh:

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date:   2015-12-24

REPOSITORY=~/Downloads/openwrt-fanqiang
CONFIG=~/Downloads/openwrt-dir505

createdir() {
    rm -rf $CONFIG
    mkdir $CONFIG
}

copy() {
    cp -R $REPOSITORY/openwrt/default/* $CONFIG/
    cp -R $REPOSITORY/openwrt/dir505/* $CONFIG/
}

setmod() {
    chmod +x $CONFIG/usr/bin/shadowsocks-firewall
    chmod +x $CONFIG/etc/uci-defaults
    chmod +x $CONFIG/etc/uci-defaults/*
```

```
}

modify() {
    # server ip address
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/etc/shadowsocks.json

    # server_port
    sed -i 's/1098/server_port/' $CONFIG/etc/shadowsocks.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/etc/shadowsocks.json

    # password
    sed -i 's/killgfw/killgfw/' $CONFIG/etc/shadowsocks.json

    # method
    sed -i 's/aes-256-cfb/aes-256-cfb/' $CONFIG/etc/shadowsocks.json


    # server ip addresss
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/usr/bin/shadowsocks-firewall

    # local_port
    sed -i 's/7654/7654/' $CONFIG/usr/bin/shadowsocks-firewall


    # ppoe username
    sed -i 's/wan-username/wan-username/' $CONFIG/usr/bin/router

    # ppoe password
    sed -i 's/wan-password/wan-password/' $CONFIG/usr/bin/router

    # wifi password
    sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/usr/bin/ap
    sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/usr/bin/router

    # root password
    sed -i 's/\\nfanqiang/\\nfanqiang/' $CONFIG/etc/uci-defaults/defaults
}


if [ "$1" = "createdir" ]; then
    createdir
elif [ "$1" = "copy" ]; then
    copy
elif [ "$1" = "setmod" ]; then
    setmod
elif [ "$1" = "modify" ]; then
    modify
else
    echo "usage: createdir copy setmod modify"
fi
```

用法：在 config-dir505.sh所在目录运行：

```
./config-dir505.sh createdir
./config-dir505.sh copy
./config-dir505.sh setmod
./config-dir505.sh modify
```

开始编译OpenWrt自动翻墙固件

```
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64
make image PROFILE=DIR505A1 PACKAGES="base-files busybox dnsmasq dropbear firewall fstools jsonfilter libc libgcc mtd netifd opkg procd swconfig ubox ubus ubusd uci usign kmod-ledtrig-usbdev kmod-lib-crc-ccitt kmod-nls-base kmod-ip6tables kmod-ipt-conntrack kmod-ipt-core kmod-ipt-nat kmod-nf-conntrack kmod-nf-conntrack6 kmod-nf-ipt kmod-nf-ipt6 kmod-nf-nat kmod-ipv6 kmod-ppp kmod-pppoe kmod-pppox kmod-slhc kmod-gpio-button-hotplug kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-ath kmod-ath9k kmod-ath9k-common kmod-cfg80211 kmod-mac80211 libip4tc libip6tc libxtables libblobmsg-json libexpat libiwinfo libjson-c libnl-tiny libubox libubus libuci ip6tables iptables hostapd-common iw odhcp6c odhcpd ppp ppp-mod-pppoe wpad-mini iwinfo jshn libjson-script uboot-envtools luci-ssl ipset wget shadowsocks-libev iptables-mod-nat-extra bind-dig" FILES=~/Downloads/openwrt-dir505
```

报告错误：

> opkg_install_cmd: Cannot install package kmod-ipv6

移除kmod-ipv6再次编译，成功。

查看编译好的固件:

```
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-generic.Linux-x86_64/bin/ar71xx/
ls -lh *505*.bin
... 7.6M Dec 24 15:20 openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin
... 5.4M Dec 24 15:20 openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin
```

升级固件要用到的是 openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin，如果在原厂固件上刷要用openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

先本地修改好配置文件再编译，然后把翻墙固件刷进D-Link DIR-505 A1后，就能零设置智能、自动翻墙。

只要配置文件设置不出差错，编译固件一般都能成功，保存好这个固件，以后随便折腾路由器，出现问题大不了重刷一次，几分钟时间就一切都恢复正常。

参考：

- http://wiki.openwrt.org/doc/howto/obtain.firmware.generate

D-Link DIR-505 A1 刷通用OpenWrt

照前面的教程自己编译翻墙固件，编译出来后刷进路由器，就能实现零设置自动翻墙。出于各种原因，有的朋友可能不想自己编译固件，又想用DIR-505实现智能翻墙，就要下载预编译的通用翻墙固件，刷好后，登录路由器，用vi修改少数几个设置，就能实现智能翻墙，本教程就是针对这些朋友而写。

路由器的开关拨到刻有 Router/AP 字样的档位，如果你没有给路由器动过手术，就是从上往下数的第一档。

DIR-505原厂固件刷翻墙固件的方法

适合购买了D-Link DIR-505 A1后没有刷过任何固件的朋友。

刷修改了16进制值的原厂固件

到下面地址下载已经修改了16进制值的原厂固件:

https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/

照官方手册说明网页登录路由器，刷新固件

刷DIR-505的翻墙固件 factory.bin

到下面地址下载用于 DIR-505的翻墙固件:

https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/

下载 openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

按照官方手册的说明刷新固件。

OpenWrt固件基础上升级到翻墙固件

注意，下面步骤适合于你已经在你的DIR-505上刷了OpenWrt固件,你想要升级到可以自己翻墙的openwrt固件。

下载翻墙固件 sysupgrade.bin

到下面地址下载用于 DIR-505的翻墙固件 openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin:

https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/

命令行上传固件到路由器

电脑通过网线或无线连接到路由器，然后: :

```
cd ~/Downloads/OpenWrt-ImageBuilder-ar71xx-nand.Linux-x86_64/bin/ar71xx/
scp openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

ssh登录OpenWrt路由器

```
ssh root@192.168.1.1
cd /tmp
```

sysupgrade升级固件并取消保留原来配置文件

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade
.bin
```

参数 `-n` 表示升级时不保留原来的配置文件。

等待两分钟等刷新固件并重启完成。

登录并设置 DIR-505 OpenWrt 翻墙固件

ADSL Modem网线连接到路由器的有线接口。路由器的开关拨到刻有 Router/AP 字样的档位，如果你没有给路由器动过手术，就是从上往下数的第一档。本文以router模式为例，如果你的应用场景是ap模式，请自行相应变通。

电脑连接DIR-505路由器

电脑连接到无线 网络 **eastking-dir505**

无线密码:

```
2014-09-01版： wsjdw,8181
新版都是： icanfly9876
```

ssh 登录 OpenWrt 翻墙固件

```
ssh root@192.168.1.1
```

输入密码 `fanqiang` 登录ssh

有时会提示错误：

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
cf:c5:12:34:56:0b:4d:1c:56:48:6a:87:04:cf:b8:83.
Please contact your system administrator.
Add correct host key in /home/openwrt-fanqiang/.ssh/known_hosts to get rid of this mes
sage.
Offending RSA key in /home/openwrt-fanqiang/.ssh/known_hosts:3
  remove with: ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
RSA host key for 192.168.1.1 has changed and you have requested strict checking.
Host key verification failed.
```

解决办法就是复制并运行提示中的清理命令：

```
ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
```

然后就可以正常登录了。

登录后用vi修改设置：

```
root@OpenWrt:~#  vi /etc/shadowsocks.json
root@OpenWrt:~#  vi /usr/bin/router
root@OpenWrt:~#  vi /usr/bin/ap     #如果是ap模式
root@OpenWrt:~#  vi /usr/bin/shadowsocks-firewall
```

分别修改以下值：

- shadowsocks.json中，server改成你的服务器实际IP
- router/ap中　wan-username 和 wan-password改成实际值
- shadowsocks-firewall中，1.0.9.8必须改成你的服务器实际IP

如果你还改了其他默认值，请自行修改相应文件。不建议修改其他默认值，以提高一次成功率。熟悉以后，建议修改shadowsock密码。

执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks restart
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/network restart

# 查看 dnsmasq ss-redir ss-tunnel是否在运行。翻墙出出现故障的时候也要查看：
ps
```

2015-12-24测试router模式， 修改配置文件，编译出固件，刷进路由器，然后不用再修改任何设置就可以翻墙。

等待约两分钟，就可以测试是否可以在网上畅行无阻了。

其他翻墙软件、方案教程

本教程主要内容是 路由器刷 OpenWrt，安装 shadowsocks翻墙。有时也要用一下其他翻墙软件。

最简单的路由器刷OpenWrt固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读OpenWrt翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

利用lantern 蓝灯实现浏览器自动翻墙教程

蓝灯运用了多种技术，通过自有服务器或者运行lantern的用户转发流量实现浏览器全自动翻墙。

latern蓝灯和 OpenWrt shadowsocks翻墙的区别

- 蓝灯主要是浏览器自动翻墙
- 路由器OpenWrt shadowsocks翻墙方案 是所有接入的设备都自动翻墙,可定制性更高

为什么选择 lantern 蓝灯翻墙

有很多的翻墙软件，有闭源的，也有开源的，我们优先选择开源软件。闭源软件缺少外界监督，不能保证没有问题。

蓝灯就是优秀的开源翻墙软件。今天是2016-01-10，在Github上已经 6516 Star, 2228 Fork，开发很活跃。

下载 lantern蓝灯翻墙软件

Github下载：
https://github.com/getlantern/lantern

主页下载:
https://getlantern.org/

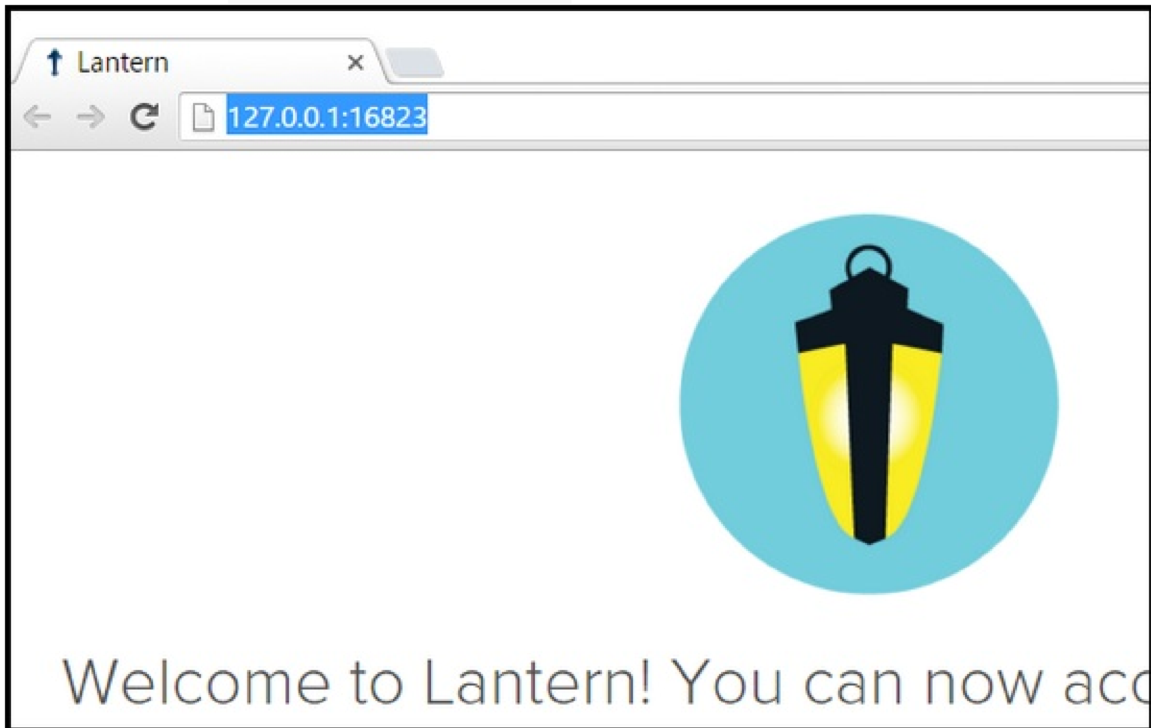
蓝灯翻墙软件安装和设置

- 停止路由器的shadowsocks翻墙
  登录OpenWrt路由器，运行命令:

  ```
  /etc/init.d/shadowsocks stop
  ```

  如果你是按照 https://github.com/softwaredownload/openwrt-fanqiang 设置的翻墙，那么还得检查一下 /etc/init.d/shadowsocks 里的start, stop函数是否正确。2016-01-10前这两个函数有bug，导致执行stop后上网不正常。

- 打开 Chrome浏览器
- 安装或运行lantern蓝灯，会自动在Chrome里打开新的页面，地址栏显示了翻墙转发的地址和端口，比如 `http://127.0.0.1:16823/`



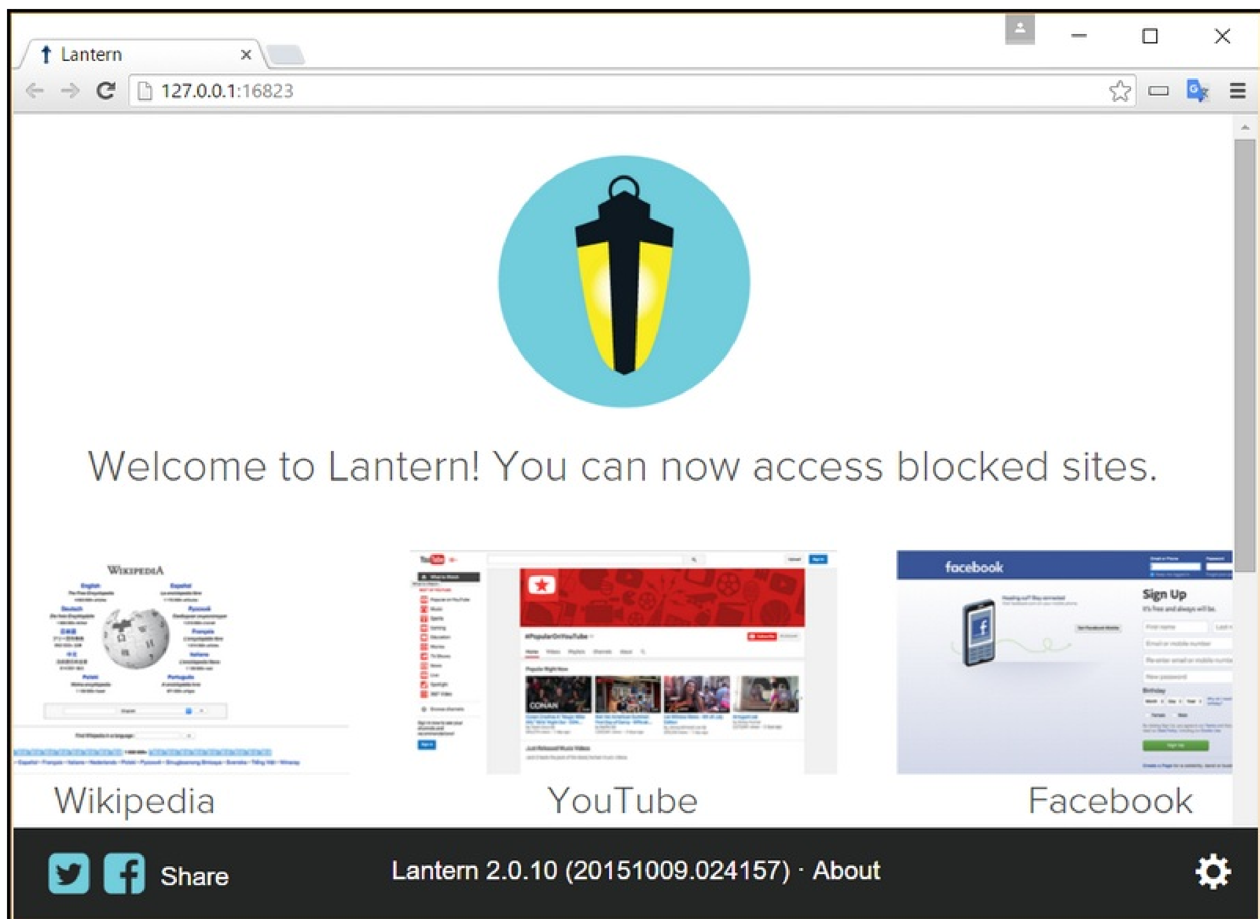
- 点击打开页面的右下角的齿轮图标设置lantern翻墙配置:



- 右键点击电脑右下角托盘图标退出lantern(Windows 为例)



如果一切正常，一运行蓝灯，就可以点击蓝灯新打开的页面上的 YouTube 图标看视频了，非常方便。



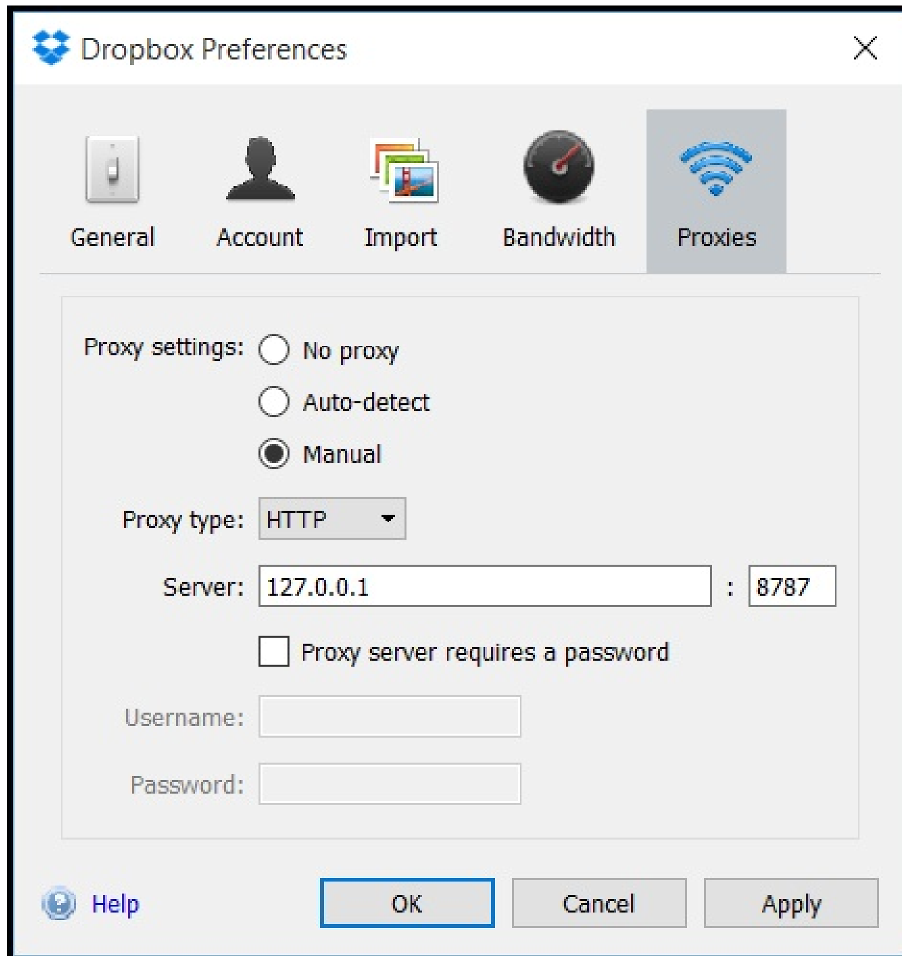
配置网络软件走 Lantern 翻墙代理:

蓝灯默认会在 127.0.0.1 上开启一个 HTTP 代理,端口号是 8787
在网络软件的代理界面上设置 HTTP 代理:

```
地址: 127.0.0.1
端口号: 8787
```

（注：“127.0.0.1”表示“本机地址”）

于是，开启Lantern, Dropbox就可以正常使用了：



lantern蓝灯翻墙软件配置文件研究

进入lantern蓝灯翻墙软件安装目录：

Windows下进入lantern安装目录

按Windows键，输入
%appdata%

然后就可以进入 Lantern 安装目录。

Mac 下进入lantern安装目录

```
/Users/name/Library/Application Support/Lantern
```

配置文件：Lantern/lantern-2.0.10.yaml

2.0.10是版本号，随不同版本而变化。

log 文件，可以了解翻墙详细过程：

```
Lantern/logs/lantern.log

...
geolookup.go:161 Successfully looked up IP '1.0.9.8' and country 'CN'
...
```

Lantern配置文件中的流量转发服务器IP地址

Lantern/lantern-2.0.10.yaml 中找到类似如下内容，替换成其他服务器，把文件设为只读，就可以更换服务器：

```
fallback-1.0.9.8:
  addr: 1.0.9.8:443
  pipelined: false
  cert: "-----BEGIN CERTIFICATE-----\n...\n-----END
    CERTIFICATE-----\n"
  authtoken: B... https://github.com/softwaredownload/openwrt-fanqiang ...C
```

Ubuntu下自己编译lantern翻墙软件

先准备好Go语言开发环境，假设Go程序的源码在 `~/golib/src` 目录下。

```
sudo apt-get update
sudo apt-get install -y git curl libappindicator3-dev build-essential libgtk-3-dev

# Use the Go compiler to build the lantern binary
cd ~/golib/src
git clone https://github.com/getlantern/lantern.git

cd lantern
source setenv.bash
go build -o lantern github.com/getlantern/flashlight

# Use curl to test that the proxy is working fine:
curl -x 127.0.0.1:8787 https://www.google.com/humans.txt

# This line will run Lantern without opening the browser window:
./lantern -headless
```

Reference:

- https://github.com/getlantern/lantern

- https://getlantern.org
- https://github.com/getlantern/lantern/blob/393657edd298268b66ad0bf3184ad7b6f21da5c0/README.md

怎样加强上网的匿名性

即使翻墙上网了，眞实的上网信息，如本机IP地址，系统语言,系统时区等等还是可能暴露。

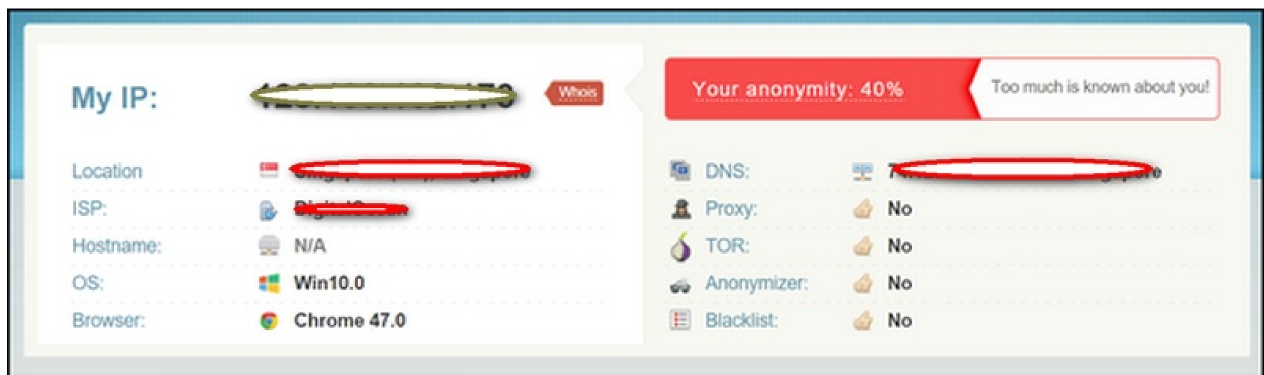
怎样检查翻墙后浏览器上网的匿名性

访问下面网站检查自己的匿名程度：

https://whoer.net/#extended

蓝灯翻墙，浏览器匿名程度测试

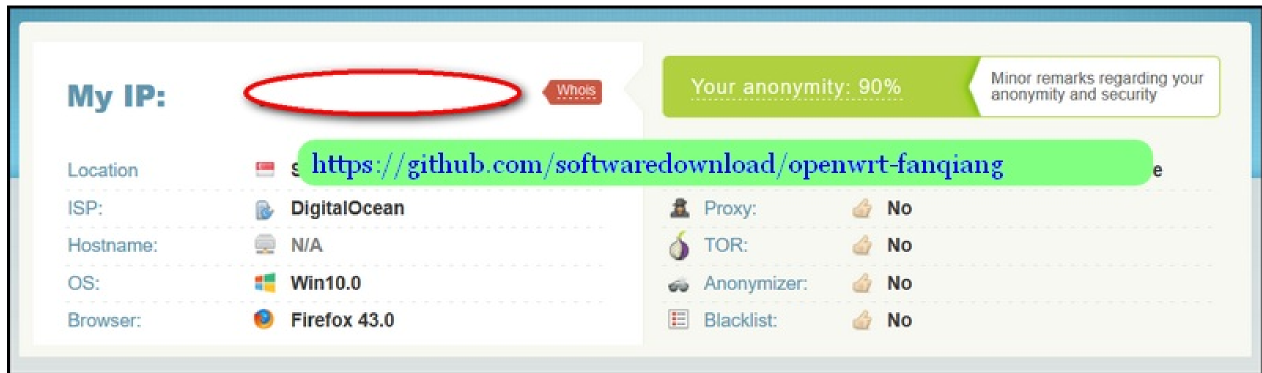
下图，蓝灯翻墙，Chrome浏览器，匿名程度 40%，很差:



再拉下去看，WebRTC暴露了本机IP地址：

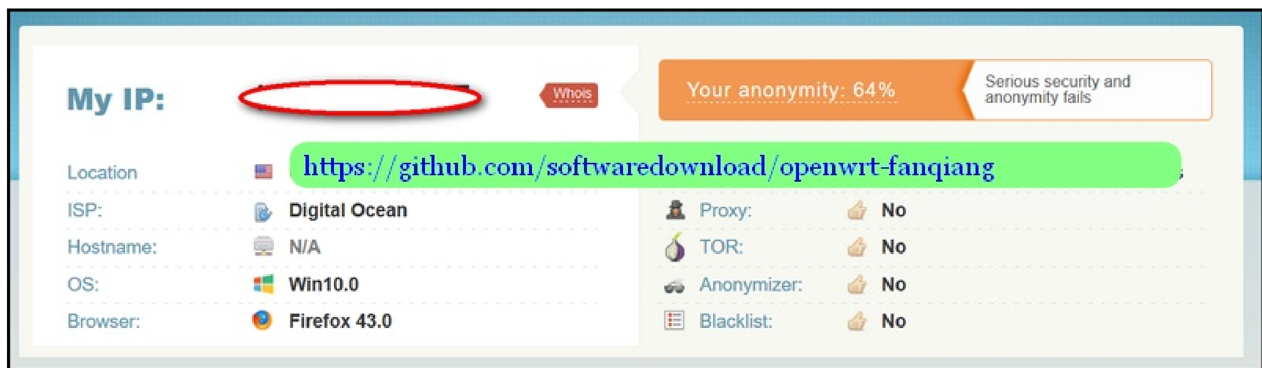


下图，蓝灯翻墙，FireFox浏览器，开启隐私设置后WeRTC已经关闭，匿名程度高达90%：

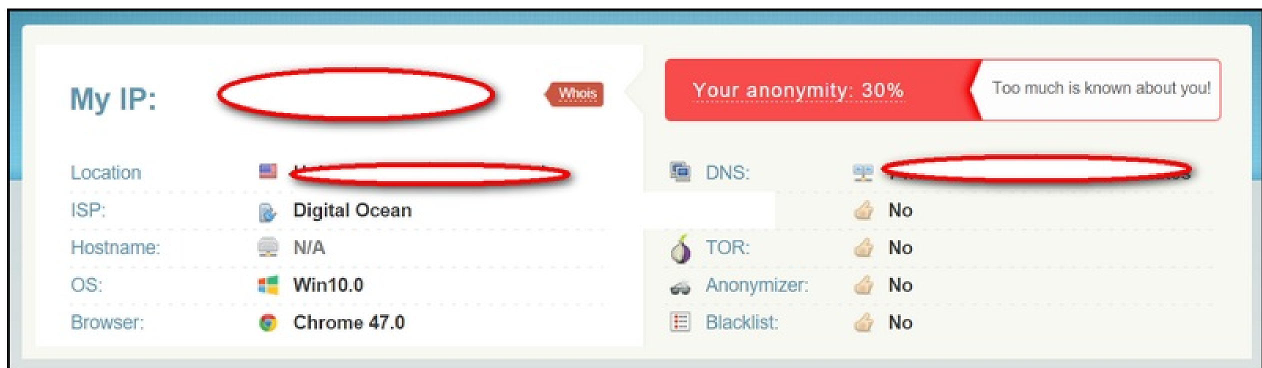


路由器刷OpenWrt，安装shadowsocks-libev翻墙，浏览器匿名程度测试

下图，FireFox浏览器，同样设置，WeRTC已经关闭，匿名程度64%：



Chrome浏览器，匿名程度只有30%了：



防止浏览器 WebRTC 泄露本机IP地址:

Chrome浏览器安装插件就可以了： WebRTC Leak Prevent

安装以后，路由器刷OpenWrt，安装shadowsocks-libev翻墙，Chrome浏览器的匿名程度提升到了64%

FireFox浏览器关闭 WebRTC

地址栏输入： `about config`
搜索： `media.peerconnection.enabled` 双击由true改为false，就可以彻底匿名了！

Opera浏览器安装插件：WebRTC Leak Prevent

什么是WebRTC What is WebRTC:

WebRTC，名称源自网页实时通信（Web Real-Time Communication）的缩写，是一个支持网页浏览器进行实时语音对话或视频对话的技术，是谷歌2010年以6820万美元收购Global IP Solutions公司而获得的一项技术。

WebRTC实现了基于网页的视频会议，标准是WHATWG 协议，目的是通过浏览器提供简单的javascript就可以达到实时通讯（Real-Time Communications (RTC)）能力。

WebRTC（Web Real-Time Communication）项目的最终目的主要是让Web开发者能够基于浏览器（Chrome\FireFox...）轻易快捷开发出丰富的实时多媒体应用，而无需下载安装任何插件，Web开发者也无需关注多媒体的数字信号处理过程，只需编写简单的Javascript程序即可实现，W3C等组织正在制定Javascript 标准API，目前是WebRTC 1.0版本，Draft状态；另外WebRTC还希望能够建立一个多互联网浏览器间健壮的实时通信的平台，形成开发者与浏览器厂商良好的生态环境。同时，Google也希望和致力于让WebRTC的技术成为HTML5标准之一，可见Google布局之深远。[1] WebRTC提供了视频会议的核心技术，包括音视频的采集、编解码、网络传输、显示等功能，并且还支持跨平台：windows，linux，mac，android。

附录

翻墙常用资源及如何贡献本项目

最简单的路由器刷OpenWrt固件翻墙教程:

https://github.com/softwaredownload/openwrt-fanqiang

在线阅读OpenWrt翻墙路由器教程:

https://www.gitbook.com/book/softwaredownload/openwrt-fanqiang/details

翻墙教程资源汇总

翻墙软件

- Shadowsocks Download
- Shadowsocks libev
- Lantern浏览器自动翻墙
- Gohop - VPN in GO lang
- Obfuscated OpenSSH Patch by zinglau
- Obfuscated OpenSSH by aligo
- Obfuscated OpenSSH by brl
- V2Ray 模块化的代理软件包
- Socks5_c 轻量级的 socks5 代理
- MProxy 最小的http代理
- GoProxy go写的隧道代理服务器
- XX-Net 接力GoAgent

翻墙方案

- 打造OpenWrt智能自动透明翻墙路由器
- FreeRouter_V2
- Autovpn for OpenWrt
- Proxy for GFW
- fqrouter, Android as router
- openwrt GFW
- dnsforwarder
- Autddvpn beta

翻墙辅助

- dnsmasq China List

翻墙教程

- FreeRouter V2完全手册PDF下载

- 在华为HG255D OpenWrt上安装和配置Shadowsocks并实现智能流量转发
- 用PDNSD + Google DNS 获得高速正确的dns解析
- 基于OpenWRT的自动翻墙路由器
- Tunlr-style DNS unblocking
- DNS unblocking using Dnsmasq and HAProxy

OpenWrt教程

- 跟hoowa学做智能路由
- 跟 UMU 一起玩 OpenWRT

本地阅读本教程的方法

git clone项目

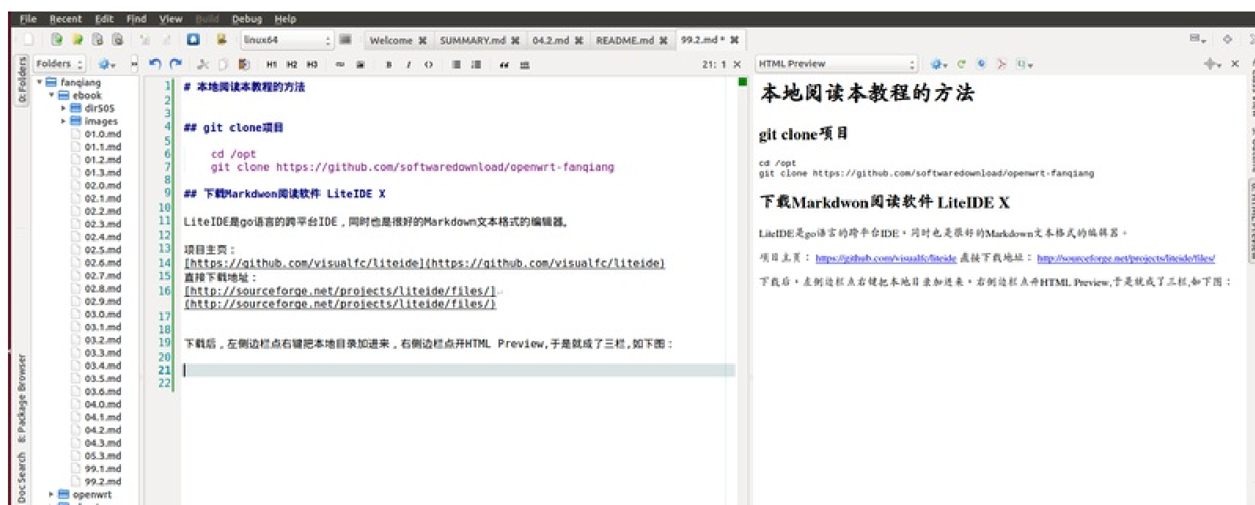
```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

下载Markdwon阅读软件 LiteIDE X

LiteIDE是go语言的跨平台IDE，同时也是很好的Markdown文本格式的编辑器。

- 项目主页： https://github.com/visualfc/liteide
- 直接下载地址： http://sourceforge.net/projects/liteide/files/

下载后，左侧边栏点右键把本地目录加进来，Alt+4 快捷键打开HTML Preview，Ubuntu中下如下图：



你是个有爱心的人，阅读了本教程，想要回馈这个开源项目，在阅读时顺便修改一些错字，加进补充内容，增加一章你的路由器应用本教程翻墙的过程等等，然后提交 pull request.

知识若不分享，实在没有意义

这个世界为什么圣人这么少？

人类历史上存在过无数人，他们都不见了，他们都到哪里去了，他们曾有过什么样的故事，可曾有人在想起他们的笑容？通过历史书，我们知道了历史上存在过的一些人物的名字，其中少数人，为人类的发展作出了特别的贡献，我们可以称他们为圣人，这样的人，一只手就数得过来。

历史上存在过的人这么多，为什么圣人却这么少？

我认为，这是因为，普通人的一生，主要是在思考怎么得到更多，而较少想到去付出。得到越多越好，付出越少越好，这就是普通人。

圣人是怎么样的，是不是只想着付出，不计收获？不是的，我认为圣人是付出得到比较均衡的人。只付出而不得到，自己就很快会陷入困境，就没有能力去帮助更多人。圣人得到什么，就会想着怎么样去回馈外界，回馈社会，在回馈过程中自己得到快速成长，从而有更大的能力去回馈更多，圣人于是逐渐长成。

我这么说，并不是希望谁成为圣人。圣人并不知道自己是圣人，也不会去想这个事情。有一个信念，就要去实行，生命的意义就在于点滴的行动，能做多少就做多少，当生命之花最终凋落时，我们得到的都将失去，我们付出的也许还会存在于这个世界很长的时间。

我为什么写这个教程

生在天朝，上网各种不方便，很是苦恼，什么OpenWrt，没有听说过，不知道哇。上网查相关论坛，非注册用户附件下载隐藏，图片隐藏，各种限制。也有一些教程散布在网上，需要自己整合。终于，花了N个白天，给家里的路由器翻墙了。我是个习惯于换位思考的人，想想自己花了很多时间查各种资料，何不花时间整合各种资源并加上自己的心得，写成系列教程，公布在网上？

于是，又是N个白天（N > 10)，学习Git, GitHub, GitBook, Ubuntu, Markdown, OpenWrt，各种调试、编译。经常一天的绝大部分时间在写这个教程。钱可以少赚些，当下够用就行，这个教程还得认真写，没有想过要得到什么，只是觉得白发已生，人生不能虚度，给这个世界留下一些自己的印记也总是好的。虽然不对别人说，但也未尝不可在人少时偷偷笑一声，并对自己说：我这样的好人，在这个世界上可是不多呢，哈哈。

为什么以开源方式发布在GitHub

为什么不写在博客上呢？如果写在博客上，就要自己维护博客，一直维护下去总是个麻烦事。GitHub总比自己维护的博客稳定，或者说能存在更长时间。即使GitHub倒闭，也就一个git命令就可以托管到其他网站，何况GitHub至少现在看来是来日方长呢。

开源方式发布，更是希望阅读本教程翻墙成功的朋友，如果你的路由器型号不被本教程覆盖，就写下自己的翻墙实践过程，提交到本项目中，以帮助相关朋友。我在教程中以 D-Link DIR-505为范例，演示了如何参与到本项目中来，将在下一节详述。

如何贡献本项目

虽然说原理是通用的，本教程内容可以应用到绝大多数路由器中去。然而，高手毕竟少数，多数有翻墙需求的人可能都没有用过Linux系统，没有听说过OpenWrt，针对他们，最好是一种路由器类型（型号）一个教程。并且最好提供预编译的固件，刷上这个预编译的固件后，修改极少的参数就可以自动翻墙。

在你应用本教程原理翻墙的过程中，把详细应用过程一步步写下来，并贡献到本项目中，以帮助更多的人。

假如你的路由器是 netgear wndr3800

如何通过 Github 贡献本项目：

先阅读 Github 贡献向导,然后：

- Fork 本项目 (https://github.com/softwaredownload/openwrt-fanqiang/fork)
- 创建你的分支 (git checkout -b my-new-feature)
- 提交你的改进 (git commit -am 'Add some feature')
- Push到你的分支 (git push origin my-new-feature)
- 到github.com 创建 Pull Request

如何为新的路由器创建翻墙教程：

```
cd openwrt-fanqiang
mkdir -p ebook/wndr3800/images
mkdir openwrt/wndr3800
```

在ebook目录下创建以路由器型号为名的目录，以wndr3800为例，教程在ebook/wndr3800目录下，图片在wndr3800/images在目录下。

wndr3800专用的配置文件在openwrt/wndr3800下，注意，openwrt/default目录已经有的配置文件可以省略。

路径、文件名都小写，因Windows系统是大小写不敏感的。

在你的教程中最好提供预编译固件的稳定下载地址。如果你没有稳定的下载空间，可以提交一个issue,附上临时下载地址，我会上传到稳定下载地址，然后你可以修改教程加上稳定下载地址。注意教程目录下不要直接包含固件文件，大的二进制文件不需要用git跟踪。

你可以用LiteIDE写教程。

修改目录文件， openwrt/SUMMARY.md，把你的教程作为新的一章，放在最后一章之前。

如果你的路由器型号与教程中的相同或类似，你也可以参与到本项目中来，你可以修正错误，补充不详细的地方，文字润色，提出建议等。